NASA Technical Memorandum 108509

Mars Global Reference Atmospheric Model (Mars-GRAM 3.34): Programmer's Guide

C.G. Justus, B.F. James, and D.L. Johnson

May 1996

NASA Technical Memorandum 108509

Mars Global Reference Atmospheric Model (Mars-GRAM 3.34): Programmer's Guide

C.G. Justus
Computer Sciences Corporation • Huntsville, Alabama

B.F. James and D.L. Johnson
Marshall Space Flight Center • MSFC, Alabama

National Aeronautics and Space Administration
Marshall Space Flight Center • MSFC, Alabama 35812

May 1996

ACKNOWLEDGMENTS

We wish to thank Pete Theisinger and Pat Esposito, NASA Jet Propulsion Lab, who suggested the need for this Mars-GRAM programmer's guide and were instrumental in securing support for this activity from NASA Headquarters, Planetary Exploration Office, Mars Orbiter Program, project 215-000-42. Work was performed by the NASA Marshall Space Flight Center, Electromagnetics and Aerospace Environments Branch, EL23. We are grateful to Belinda Hardin for her expert assistance in preparing this report and to Margaret Alexander for her skillful editing of the draft.

PREFACE

The effort required for the preparation of this report was sponsored by the Mars Global Surveyor Project, through NASA Jet Propulsion Laboratory (Sam Dallas JPL Mission Manager), under project 215-000-042. Technical questions on the Mars-GRAM may be addressed to Dr. C. G. Justus, EL23/CSC, NASA Marshall Space Flight Center, Huntsville, Alabama 35812 (205-544-3260; e-mail jere@profiles.msfc.nasa.gov).

iv

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

vi

LIST OF TABLES

vii

Page
viii
Blank

TECHNICAL MEMORANDUM

Mars Global Reference Atmospheric Model (Mars-GRAM Version 3.34): Programmer's Guide

1. INTRODUCTION

1.1 Background

The Mars Global Reference Atmospheric Model (Mars-GRAM) was developed (Johnson et al., 1989; Justus 1990, 1991; Appendix B, version 2.21) as an engineering-oriented, empirical model of the Mars atmosphere. The model is based on surface and atmospheric temperature data observed during the Mariner and Viking (orbiter and lander) missions and on surface pressure data observed by Viking landers. At higher altitudes (above about 120 km), Mars-GRAM is based on the Stewart (1987) thermospheric model given in Pitts et al., (1990?). The model provides both mean and mountain-wave perturbed atmospheric density for any location (height, latitude, longitude) and time (seasonal, diurnal). Other atmospheric variables include atmospheric temperature, pressure, and wind components. Dust storm effects, controlled by user-selected options, are provided for the atmospheric parameters.

A second official release of Mars-GRAM (James and Justus, 1993; Appendix C, version 3.1) added several new capabilities, e.g., the option to simulate either local-scale or global-scale dust storms and density perturbations from tidal waves by the Zurek wave model given in Pitts et al. (1990?).

Table 1-1 gives a brief history of Mars-GRAM program development through the current version — 3.34. Newest features include (1) a limitation based on atmospheric stability considerations for magnitude of mountain-wave density perturbations, (2) comparisons of density, temperature, and pressure with the COSPAR reference atmosphere (Pitts et al., 1990?), and (3) a new method for estimating the diurnal range of surface temperature based on diurnal variability of surface-absorbed solar energy. Technical descriptions of these new features are discussed in Section 3.0.

Table 1-1. History of Mars-GRAM Program Versions (Page 1 of 2)

Version	Date	Comments
1.00	5/20/88	Preliminary version with earlier Stewart thermosphere model and no realistic latitude-longitude variation. Documented in ED44-5-20-88 preliminary report.
2.00	7/1/89	Version documented in July, 1989 technical report. Has newer Stewart thermosphere model and realistic latitude-longitude variability.
2.10	9/2/89	Adds version numbers to main and all subroutines. Corrects formats 790 and 795 in DATASTEP subroutine.
2.11	10/2/89	Corrects "ATIO" to "RATIO" and "FH" to "PFH" in THERMOS subroutine. Corrects MARSGRAM, ATMOS2, DATASTEP, PRESSURE, PSURFACE, STEWART2, STRATOS, TEMPS, and TSURFACE to have lines <= 72 characters in length.
2.20	10/7/89	Corrects illegal log call in THERMOS by adding ES factors to ZF in ATMOS2. Adds EScalc subroutine in STEWART2. Changes name of terrain height file to HEIGHTS.DAT
2.21	10/8/89	Removes character data from COMMON DATA and puts it in new COMMON FILENAME.
2.22	11/16/89	Adds REAL J2 to RELLIPS, and REAL nmals to PSURFACE. Changes Julian date by -0.5 to be consistent with astronomical convention of day starting at Greenwich noon. ORBIT adds back 0.5 to Julian day for consistency with derivation of coefficients.
3.0	10/14/91	Adds option for local-scale dust storm, and Zurek wave perturbation model. Allows heights to go "below" local terrain height and return "realistic" pressure, density and temperature, not the surface values. Both Interactive and Batch versions available. Batch version uses NAMELIST input, and is completely modular, so main driver program can easily be replaced by any calling program, such as a trajectory simulation program.
3.1	12/17/92	Change comments and code for DENSRP output to perturbations in %; change DENSLO output file to DENSRM, containing random perturbation magnitude in %; change DENSHI output file to DENSWA, containing wave perturbation in %; add DENSWA to the OUTPUT file. Modify DENSHI and DENSLO to include the wave perturbation amplitude. Delete several unused variables from declaration statements in MAIN routine and subroutines SETUP, ATMOS2, PSURFACE, STEWART2, DZDUST, THERMOS, and STRATOS.
3.1	3/14/94	Transferred version 3.1 to MSFC Unix environment and tested.
3.2	11/28/94	Corrected DENSHI, DENSLO initial value problem. Added parameter value output to DATASTEP. Modified DZDUST to use same dust storm start time and intensity as lower altitude. Changed output file units to iu0 for messages (normally screen) and iup for LIST output (iup = 0 also suppresses LIST and other output in batch version).

Table 1-1. History of Mars-GRAM Program Versions (Page 2 of 2)

Version	Date	Comments
		Corrected error in calculation of time-step correlations. Added line number codes in columns 73-80. Sorted MARSSUBS into alphabetical order by line number codes. Renamed Commons RAND to RANDCOM and DATA to DATACOM. Modified E-format in LIST output to have a leading digit. Made code consistent for indentation of If..Then..Else segments and Do..Continue loops. Moved NAMELIST read into SETUP for batch version and removed commons from the batch main (to simplify use of SETUP and DATASTEP as subroutines in other driver programs). Deleted additional unused variables in ATMOS2 and ORBIT. Defined pi180 in DATASTEP and corrected NVAR to NVARX. Corrected slight inconsistency in pressure interpolation in subroutine STRATOS. Added time (rel. to initial time) in sols to standard LIST file. Added time and solar longitude to OUTPUT list.
3.3	2/7/95	Simplified vertical interpolation method in subroutine STRATOS. Added DENSLO and DENSHI output files back ("1 sigma" density envelopes). Corrected erroneous comments associated with file open statements in batch version code. Added maxfiles option to suppress output of TMAX, TMIN and TAVG files for systems that cannot have more than 16 files open at one time (i.e. set maxfiles = 16 in Block Data routine).
3.31	3/28/95	Added check for density perturbations not to exceed value set by wave instability (i.e. maximum perturbation magnitude consistent with d-theta/dz > 0). Ensured wave amplitudes are zero if wave perturbation model is not selected. Added output of F10.7 at 1AU and Mars orbit.
3.32	4/11/95	Added COSPAR NH mean atmosphere and output of density deviations from COSPAR values (also new option logscale =2). Suppress output of NSWIND and EWWIND if maxfiles = 16.
3.33	8/9/95	Takes polar cap radius from Alb function and makes it a separate function, polecap. Uses new regressions for Tavg versus Absorb (the daily average surface-absorbed solar flux). Uses new regression for diurnal temperature amplitude (Tamp) versus Qa [the diurnal range (midnight-noon difference) in surface-absorbed solar flux]. These changes in temperature regressions are all in the Tsurface subroutine.
3.34	11/1/95	Added comments for JPL Programmer's Guide. Corrected logic error in density to OUTPUT file (added format 810). Corrected problem with crlat and polar in Tsurface. Corrected semi-diurnal term in dusty-case wave model. Corrected term in maximum mountain-wave perturbation model.

Page
1-4
Blank

2. OVERVIEW OF THE MARS-GRAM PROGRAM

The Mars-GRAM program consists of four FORTRAN source code files:

(1) The Interactive form main program is MARSGRAM.FOR (or marsgram.f on UNIX machines)

(2) The Batch (subroutine) form is MARSGRMB.FOR (or marsgrmb.f)

(3) The Setup subroutine, used only in the batch form, is SETUP.FOR (or setup.f)

(4) All other subroutines are MARSSUBS.FOR (or marssubs.f).

Throughout this report, names of programs, subroutines, and variables appear either as all-capitals or as upper/lower case names. The FORTRAN compiler option used is assumed to be case-insensitive.

Since version 3.2, all program code now has line numbers in columns 73-80. Line numbers consist of a four-character code (MARS for the interactive main, MARB for the batch main, SETU for the setup subroutine, etc.) and a three-digit number. For code inserted since version 3.2, an additional single letter appears in column 80.

Table 2-1 gives a simplified "map" of the Mars-GRAM programs by indicating names of subroutines and functions, 4-character code used on source file line numbers, and calling subroutine(s). Appendix A gives complete descriptions of the interactive and batch main programs, setup subroutine, and other subroutines. Except for the setup subroutine, all functions and subroutines are listed in Appendix A in alphabetic order by 4-character code.

Many program variables are passed between routines via "common" blocks. Table 2-2 lists the seven common blocks and subroutines that use them. Descriptions in Appendix A include definitions of all input, output and local variables, except those pass through these common blocks. Table 2-3 lists variables in the common blocks, including common block name, variable name, variable type, and description of the variable.

The interactive form of the program prompts the user for all input (with exception of data files HEIGHTS.DAT and COSPAR.DAT). The batch form reads and loads all data by the setup subroutine. The combination of setup routine and subroutines in marssubs.f are particularly suitable for adaptation as subroutines in user programs (e.g., orbital propagator and trajectory dynamics programs). See discussion of the batch form (MARB) in Appendix A and in Section 4.

Both the interactive and batch forms are essentially drivers for the DATASTEP (and in turn the ATMOS2) subroutine (details in Appendix A). A brief outline of the process steps in Mars-GRAM subroutines follows:

DATASTEP

1. Evaluate terrain height via TERRAIN subroutine

2. Obtain orbital and solar positions via ORBIT subroutine

3. Compute dust storm factors via DUSTFACT subroutine

4. Obtain pressure gradients for winds: call ATMOS2 at the current location ± 2.5° latitude and longitude

ATMOS2

5. Calculate surface temperature via TSURFACE subroutine

6. Compute lapse rates of atmospheric temperature via GAMMA subroutine

7. Evaluate temperatures at "significant levels" (0, 5, 15, 30, 50 and 75 km) via TEMPS subroutine

8. Obtain surface pressure via PSURFACE routine

9. Calculate atmospheric pressure at significant levels via PRESSURE routine

   i.e., if desired height is:

10. Below 75 km, interpolate between significant levels

11. Above the base of the thermosphere, use STEWART2

12. Between 75 km and the base of the thermosphere, interpolate via STRATOS subroutine

DATASTEP

13. Use pressure gradients, density, and Coriolis parameter to obtain wind components by the areostrophic (thermal wind) relations

14. Apply viscous damping correction to areostrophic wind components

15. Compute random (mountain wave) perturbations in density (maintain proper correlation with previous perturbation value)

16. Verify that random perturbation does not exceed stability limit

17. Compute Zurek (tidal) wave model density perturbation

18. Add perturbations to mean value of selected perturbation model.

Table 2-1. Map of Mars-GRAM Programs and Names of Subroutines, Functions, 4-character Code, and Calling Subroutines

Subroutine Name	Code	Called By
ATMOS2	ATM2	Datastep
Block Data	BLKD	N/A
cospar	COSP	Datastep
Dustfact	DSTF	Datastep
Datastep	DSTP	MarsGRAM(Main), Marsgrmb(Main)
DZDUST	DZDS	ATMOS2, STEWART2
EScalc	ESCL	ATMOS2, STEWART2
gamma	GAMA	ATMOS2
orbit	ORBT	MarsGRAM(Main), Datastep, SETUP
Pressure	PRES	ATMOS2
PRSEAS	PRSE	ATMOS2, Psurface, STEWART2
Psurface	PSRF	ATMOS2
RELLIPS	RLPS	MarsGRAM(Main), ATMOS2, Datastep, Psurface, STEWART2, THERMOS, SETUP
SETUP	SETU	Marsgrmb(Main)
Stratos	STRA	ATMOS2
STEWART2	STW2	ATMOS2
THERMOS	THRM	STEWART2
Temps	TMPS	ATMOS2
Tsurface	TSRF	ATMOS2
Wavepert	WAVE	Datastep

Function Name	Code	Called By
Alb	ALBL	Tsurface
ampint	AMPN	Wavepert
Cp	CPOT	ATMOS2
phasint	PHSN	Wavepert
polecap	POLC	Alb, Tsurface
PPND	PPND	Datastep
Random	RAND	MarsGRAM(Main), Datastep, SETUP
Tdiurnal	TDIR	Tsurface
Terrain	TERN	MarsGRAM(Main), Datastep, SETUP

Table 2-2. Common Blocks and Subroutines

Common	Used By
cosparnh	MarsGRAM, cospar, Datastep, SETUP
DATACOM	MarsGRAM, Block Data, Dustfact, Datastep, SETUP
FILENAME	MarsGRAM, Datastep, SETUP
RANDCOM	MarsGRAM, Random, SETUP
TERHGT	MarsGRAM, Terrain, SETUP
THERM	MarsGRAM, ATMOS2, STEWART2, SETUP
WAVEDAT	MarsGRAM, Block Data, Wavepert

Table 2-3. Variables in the Common Blocks (Page 1 of 2)

Variable	Type	Description
COSPARNH		
zc(159)	REAL*4	COSPAR heights (km)
tc(159)	REAL*4	COSPAR temperatures (K)
pc(159)	REAL*4	COSPAR pressures
dc(159)	REAL*4	COSPAR densities
DATACOM		
DTR	REAL*4	factor for degrees to radians
BETA	REAL*4	parameter for computing viscosity of CO2
SVAL	REAL*4	parameter for computing viscosity of CO2
DAY	REAL*4	length of sidereal day (hours)
CORFAC	REAL*4	Coriolis factor for computing winds
NPOS	INTEGER*4	maximum number of positions
NVARX	INTEGER*4	x-code for plotable output (see Table 2-3)
NVARY	INTEGER*4	y-code for plotable output (see Table 2-3)
logscale	INTEGER*4	units for pressure and density output (0 = MKS, 1= log-base-10, 2=% deviation from COSPAR)
dustlat	REAL*4	latitude of local scale dust storm
dustlon	REAL*4	longitude of local scale dust storm
dusthgt	REAL*4	height of local scale dust storm
radmax	REAL*4	maximum radius of local scale dust storm
Rref	REAL*4	Mars radius (km) of Cain 6.1 mb reference ellipse
modpert	INTEGER*4	perturbation model number
als0	REAL*4	initial areocentric longitude of sun
intens	REAL*4	dust storm intensity value
iu0	INTEGER*4	unit number for screen output
iup	INTEGER*4	unit number for LIST output (set=0 for no LIST)
maxfiles	INTEGER*4	maximum number of files that compiler can open
FILENAME		
lstfl	CHAR*12	name of LIST file
outfl	CHAR*12	name of OUTPUT file
RANDCOM		
IX	INTEGER*4	intermediate variable for random numbers
IY	INTEGER*4	intermediate variable for random numbers
IZ	INTEGER*4	intermediate variable for random numbers
TERHGT		
th(19,19)	REAL*4	terrain height versus latitude and longitude
THERM		
F107	REAL*4	10.7 cm solar flux at Earth position (1AU)
stdl	REAL*4	number of std. dev. for thermospheric variation

Table 2.3. Variables in the Common Blocks (Page 2 of 2)

WAVEDAT		
ampc100(12)	REAL*4	diurnal wave amplitude vs height, clear, lat=00
phc100(12)	REAL*4	diurnal wave phase vs height, clear, lat=00
ampd100(12)	REAL*4	diurnal wave amplitude vs height, dusty, lat=00
phd100(12)	REAL*4	diurnal wave phase vs height, dusty, lat=00
ampd200(12)	REAL*4	semidiurnal wave amplitude vs height, clear, lat=00
phd200(12)	REAL*4	semidiurnal wave phase vs height, clear, lat=00
ampc120(12)	REAL*4	diurnal wave amplitude vs height, clear, lat=20
phc120(12)	REAL*4	diurnal wave phase vs height, clear, lat=20
ampd120(12)	REAL*4	diurnal wave amplitude vs height, dusty, lat=20
phd120(12)	REAL*4	diurnal wave phase vs height, dusty, lat=20
ampd220(12)	REAL*4	semidiurnal wave amplitude vs height, clear, lat=20
phd220(12)	REAL*4	semidiurnal wave phase vs height, clear, lat=20
ampc145(12)	REAL*4	diurnal wave amplitude vs height, clear, lat=45
phc145(12)	REAL*4	diurnal wave phase vs height, clear, lat=45
ampd145(12)	REAL*4	diurnal wave amplitude vs height, dusty, lat=45
phd145(12)	REAL*4	diurnal wave phase vs height, dusty, lat=45
ampd245(12)	REAL*4	semidiurnal wave amplitude vs height, clear, lat=45
phd245(12)	REAL*4	semidiurnal wave phase vs height, clear, lat=45

Page

2-8

Blank

3. NEW MARS-GRAM FEATURES

3.1 Stability-Limited Mountain Wave Perturbations

A mountain-wave density perturbation model has been part of Mars-GRAM from the beginning. Recently noted is that effects of atmospheric stability should limit amount of growth with height attained from the mountain wave model.

Deviation of the convective stability limit on density perturbations incorporated into Mars-GRAM 3.34 is as follows:

Let the total atmospheric density be $\rho = \rho_0 + \rho'$, where ρ_0 is the mean value and ρ' is the mountain-wave perturbed value. Static instability (density overturning) would result if the density were to increase with height. Thus stability requires that

$$d\rho/dz = d\rho_0/dz + d\rho'/dz < 0 \quad , \tag{1}$$

or

$$d\rho'/dz < -d\rho_0/dz = \rho_0/H \quad , \tag{2}$$

where H is the scale height. If we approximate $(d\rho'/dz)$ as $2\ \rho'_{max}/L$, where L is the vertical scale (or wavelength), then the stability limit for density overturning is

$$\rho'_{max}/\rho_0 < L/(2\ H) \quad . \tag{3}$$

A more stringent stability condition (usually) than density overturning is the convective stability limit that temperature lapse rate (-dT/dz) not exceed adiabatic lapse rate (g/Cp, where g = gravity, Cp = specific heat at constant pressure). Thus, the convective stability constraint means that

$$-(dT_0/dz + dT'/dz) < g/Cp \quad , \tag{4}$$

or

$$-dT'/dz < (g/Cp + dT_0/dz) \quad . \tag{5}$$

If we approximate $-dT'/dz$ as $2\,T'_{max} / L$, the convective stability limitation then becomes

$$T'_{max} / T_0 \; < \; (g/Cp \; + \; dT_0/dz)\, L / (2\, T_0) \quad , \tag{6}$$

or, in terms of the Brunt-Vaisala frequency, ω_B,

$$T'_{max} / T_0 \; < \; \omega_B^{\,2}\, L / (2\, g) \quad . \tag{7}$$

If first order versions of the perturbed perfect gas law and hydrostatic balance relations,

$$p' / p_0 \; = \; \rho' / \rho_0 \; + \; T' / T_0 \quad , \tag{8}$$

and

$$d\,(p' / p_0) / dz \; = \; (\,1 / H\,)\,(T' / T_0\,) \tag{9}$$

are used to convert the temperature perturbation in equation (7) to a density perturbation. The convective stability limit on density perturbations then becomes

$$\rho' / \rho_0 \; < \; [\, \omega_B^{\,2}\, L / (2\, g)\,]\,[\,1 \; + \; L / (2\, H)\,] \; . \tag{10}$$

This convective stability limit, via equation (10), on allowable magnitude for mountain-wave perturbations has now been incorporated into the subroutine DATASTEP (DSTP199c), with a constraint that the density overturning condition of equation (3) also be met (DSTP199d). See further discussion of the DATASTEP subroutine in Appendix A.

3.2 Comparison to COSPAR Reference Atmosphere

Mars-GRAM Version 3.34 includes a comparison of model values of density, temperature, and pressure with those of the COSPAR reference atmosphere (Table XI of Pitts, et al., 1990?). COSPAR data values are read in as a height array, from a file named COSPAR.DAT. COSPAR values for a desired height are found by interpolating between heights in the tabular data. See discussion of the subroutine COSPAR in Appendix A for technical details of the methodology.

3.3 New Surface Temperature Parameterizations

In early versions of Mars-GRAM, surface temperatures were estimated (TSURFACE subroutine) by empirically-derived regressions of temperature versus A, the daily average amount of solar radiation absorbed at the surface. Separate regressions were used for daily minimum and daily maximum temperature, via

$$T_{min} = a_{min} + b_{min} A + c_{min} A^2 \tag{11}$$

and

$$T_{max} = a_{max} + b_{max} A + c_{max} A^2 \quad , \tag{12}$$

with average daily temperature of $(T_{min} + T_{max})/2$.

A recent re-analysis of original Viking lander and InfraRed Thermal Mapper (IRTM) data, yielded an improved approach for estimating surface temperatures. First, an improved method is used to compute surface absorption, A, via

$$A = < \tau > (1 - a) F_0 < \cos(\theta) > \quad , \tag{13}$$

where $< \tau >$ is the daily average solar transmittance, a is the surface albedo (from ALB function), F_0 is the top-of-atmosphere, direct-normal solar flux (for given latitude and day), and $<\cos(\theta)>$ is the daily average of cosine of solar zenith angle, θ. The average transmittance is computed by methods of Justus and Paris (1985), via

$$< \tau > = \varpi / 2 + (1 - \varpi / 2) \exp(-\delta / \mu_0) \quad , \tag{14}$$

where ϖ is the single-scatter albedo of the dust (taken to be 0.85), δ is the optical depth of the dust (taken to be 0.3), and μ_0 is the cosine of the noontime solar zenith angle, given by

$$\mu_0 = \sin(\varphi) \sin(\varphi_s) + \cos(\varphi) \cos(\varphi_s) \quad , \tag{15}$$

where φ is local latitude and φ_s is latitude of the Sun.

The simplified relation of equation (14) was compared with the results of a subroutine (FFACT) developed by Davies (1979) from accurate Monte Carlo radiative transfer calculations. Equation (14) results were found to agree with FFACT values within a root-mean-square value of about 0.03, or roughly the same accuracy level as FFACT values reproduce original Monte Carlo simulations (Davies, 1979).

With surface absorption, A, determined from equation (13), new regression relations were derived for T_{avg}, the daily average temperature via

$$T_{avg} = a + b\,A + c\,A^2 \quad , \tag{16}$$

if the latitude is outside the polar cap boundary, and

$$T_{avg} = a_{cap} + b_{cap}\,A - c_{cap}\,P \quad , \tag{17}$$

if latitude is inside the polar cap boundary. P is a polar cap correction that varies from 0 at the polar cap boundary to a maximum of 1 at the pole, when the cap boundary is at its largest seasonal value (factor polar in TSURFACE, line TSRF 35, Appendix A).

New methodology for surface temperature estimates assumes that the daily range of surface temperatures, $T_{amp} = (T_{max} - T_{min})$, is proportional to the daily range of surface absorption, Q, where Q is given by

$$Q = <\tau> (1 - a)\,F_0\,[<\cos(\theta)> - \mu_n] / 2 \quad , \tag{18}$$

where the symbolism is the same as for equation (13) and μ_n is the midnight solar zenith angle, via

$$\mu_n = \sin(\varphi)\sin(\varphi_s) - \cos(\varphi)\cos(\varphi_s) \quad , \tag{19}$$

for latitudes at which the Sun is above the horizon at midnight; otherwise, $\mu_n = 0$.

The daily surface temperature range is determined from

$$T_{amp} = 0.16\,Q \tag{20}$$

and daily minimum and maximum surface temperatures are given by

$$T_{min} = T_{avg} - T_{amp} \tag{21}$$

$$T_{max} = T_{avg} + T_{amp} \quad . \tag{22}$$

For additional technical description of the new surface temperature methodology, see subroutine TSURFACE in Appendix A.

The new regressions make relatively little difference in the average surface temperature. However, for cases in which the daily average absorption, A, is relatively large while the daily range in surface absorption, Q, is small, a significant change (reduction in daily temperature range) from the previous regressions can occur. This effect, illustrated in Figure 1 is most significant at high northern latitudes during northern hemisphere summer (near $L_S = 90°$) and at high southern latitudes during southern hemisphere summer (near $L_S = 270°$). Changes in these seasons and latitude ranges would be apparent in revised plots of surface temperature analogous to Figures 1, 2, and 3 of Appendix B. The reduced diurnal range in temperatures that results from the new regressions is considered to be significantly more realistic in these cases.

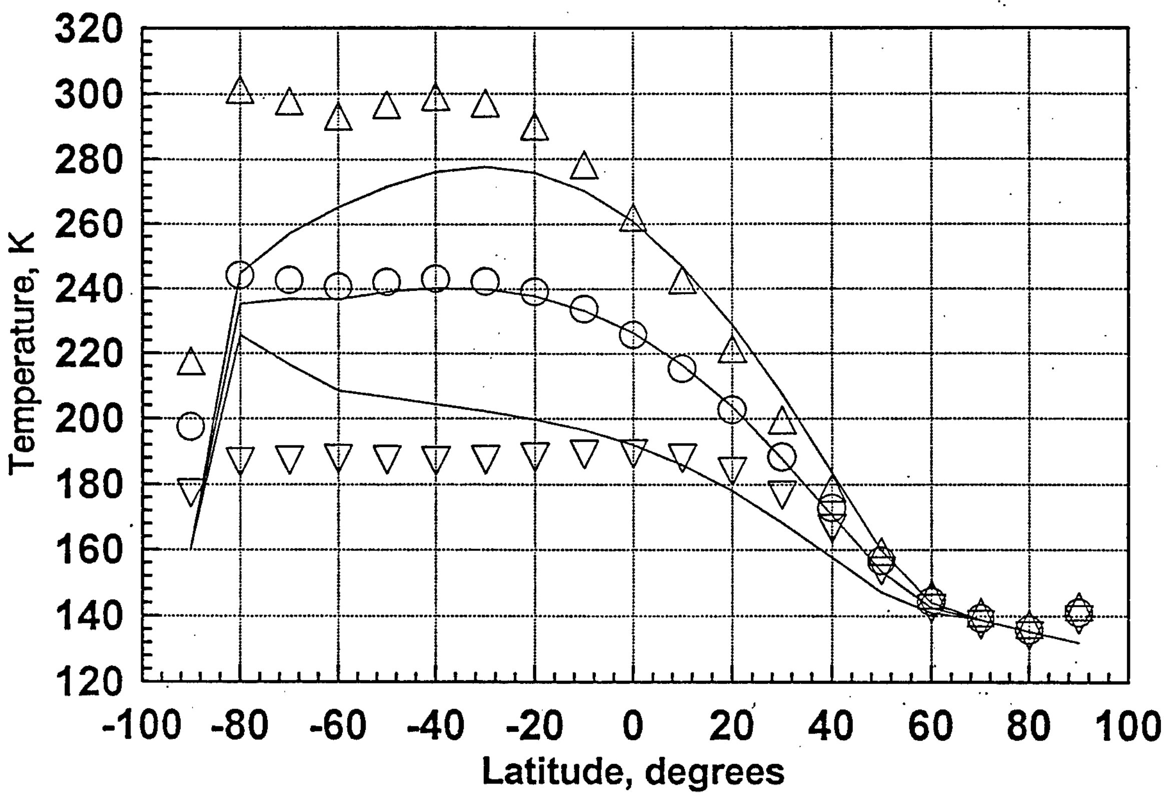


Figure 3-1. Comparison of Old and New Parameterizations for Surface Temperature in Mars-GRAM. The season is southern hemisphere summer (L_s=270°). Lines are the new regression values for minimum, average and maximum daily surface temperature. Symbols are for the old regression results.

4. HOW TO RUN MARS-GRAM

4.1 Program Input

Two forms of Mars-GRAM are: interactive, in which values for all input options are provided interactively by the user at run time, and batch, in which values for all input options are provided by a NAMELIST format input file. Table 4-1 illustrates a sample operation of the interactive form of Mars-GRAM version 3.34 and Table 4-2 gives a sample of the NAMELIST file INPUT for the batch form of Mars-GRAM 3.34.

For both the interactive or batch forms, values of the following input variables must be supplied:

LSTFL	Name of LIST file (see Table 4-4). For a listing to the console in interactive form enter filename CON.
OUTFL	Name of OUTPUT file (see Table 4-5).
MONTH	Month (1-12) for initial time
MDAY	Day of the month for initial time
MYEAR	Year for starting time is a 4-digit number. Alternative: input years 1970-2069 as a 2-digit number
NPOS	Maximum number of positions to evaluate to automatically generate a profile. Use 0 if trajectory positions are read in from a TRAJDATA file.
IHR	Initial time, hour of the day GMT
IMIN	Initial time, minute of the hour
SEC	Initial time, second of the minute
ALS0	Value of the areocentric longitude of the Sun (Ls, in degrees) where a dust storm is to start. Use 0 if no dust storm is to be simulated. (Dust storm can be simulated only during the season of the Mars year for Ls between 180 and 320 degrees.)
INTENS	Dust storm intensity, an arbitrary intensity scale, with allowable values from 0.0 (no dust storm) to 3.0 (maximum intensity dust storm). [No prompt occurs in interactive form, if no dust storm is to be simulated (ALS0=0).]

RADMAX	Maximum radius (km) a dust storm attains, developing according to the parameterized space and time profile of build up and decay in the program. If 0 or >10000 km is used, the storm is considered of global dimensions (uniformly covering the planet), but assumed to build up and decay in intensity according to the same temporal profile. [No prompt occurs in interactive, if no dust storm is to be simulated.]
DUSTLAT	Latitude (degrees, North positive) for center of dust storm. [No prompt occurs in interactive, if no dust storm is to be simulated or storm has global dimensions.]
DUSTLON	Longitude (degrees, West positive) for center of dust storm. [No prompt occurs in interactive, if no dust storm is to be simulated or storm has global dimensions.]
F107	10.7-cm solar flux (units 10^{-22} W/cm^2) at the average Earth orbit position (1 AU). Program automatically converts solar flux to its value at orbit position of Mars.
STDL	Standard deviation parameter for long-term variations in the Stewart model thermosphere. Normal value is 0; allowable range is -3.0 to +3.0.
MODPERT	Model number for perturbations to be computed: 1 is for random (mountain wave) model , 2 is for Zurek (tidal) wave model, and 3 means use combined perturbations from both models.
NR1	Seed value (integer) for random number generator. Allowable range is 1 to 29999. No prompt occurs in interactive if MODPERT = 2. To do Monte Carlo simulations with a variety of perturbations, use a different random number seed on each run. To repeat a given perturbation sequence on a later run, use same random number seed value.
NVARX	x-code for plotable output (x-y pairs for 1-D line graphs or x-y-z triplets for 2-D contour plots). See Table 4-3 for list of variables associated with x-code (e.g., if NVARX = 1, output is for plotting versus height above reference ellipsoid).
NVARY	y-code for 2-D contour plot output (x-y-z triplets). Use y-code 0 for 1-D line graph (x-y pair) plots. See Table 4-3 for list of y-code values and parameters.
LOGSCALE -	Parameter controls units of output values for density and pressure on output plot files. Value 0 means use regular density and pressure units

(kg/m^3 and N/m^2); 1 means output logarithm (base 10) in regular units; and 2 means output percent deviation from COSPAR values.

FLAT	Latitude of initial point to simulate (degrees, North positive)
FLON	Longitude of initial point to simulate (degrees, West positive)
FHGT	Height (km) of initial point to simulate, above reference ellipsoid
DELHGT	Height increment (km) between successive steps in automatically generated profile (positive upward)
DELLAT	Latitude increment (degrees, Northward positive) between successive steps in automatically generated profile
DELLON	Longitude increment (degrees, Westward positive) between successive steps in automatically generated profile
DELTIME	Time increment (seconds) between steps in automatically generated profile

Two auxiliary input files are also required. File HEIGHTS.DAT contains terrain height data array (terrain height, km, above the reference ellipsoid - see explanation in description of subroutine TERRAIN in Appendix A). File COSPAR.DAT contains height profile of COSPAR temperature, density, and pressure values (see explanation in description of COSPAR subroutine in Appendix A).

If the pre-computed trajectory mode is used (NPOS=0), read trajectory data from TRAJDATA file. Each line of TRAJDATA file is a position and time to compute atmospheric parameters. Input lines contain time (seconds, from initial time), height (km, relative to reference ellipsoid), latitude (degrees, North positive), and longitude (degrees, West positive).

For automatically-generated profiles, output continues until the maximum number of positions (NPOS) is reached. For trajectory positions, enter input from TRAJDATA file, output continues until end of the file is reached. For interactive, the program prompts for additional input values for initial date and number of positions. The program is terminated by giving values of 0 for requested input. (See end of Table 4-1).

Table 4-1. Sample Operation of the Interactive Form of Mars-GRAM 3.34

```
Mars-GRAM Interactive version 3.34 - November 1, 1995
Enter name for LIST file (CON for console listing):
VIKING1.LST
Enter name for OUTPUT file:
VIKING1.OUT
Enter Month, Day of Month, 4-digit Year, and Max Number Positions
7 20 76 21
Enter initial GMT Time (Hours, Minutes, Seconds)
12 30 0
Ls =   97.0 degrees for this date.
Dust storms can occur between Ls = 180 and Ls = 320.
Enter starting Ls value for dust storm (or 0 for none).
0
Enter mean F10.7 flux at 1AU (nominal value = 150)
and +/- number of std. deviations for thermosphere variation
185 0
Enter perturbation model: 1=random, 2=wave, 3=both
3
Enter Starting Random Number (any positive integer < 30,000)
1001
Select x-code and y-code for plotable output versus desired parameter(s):

Code              Parameter
----   -----------------------------------------------
  1    Height (above reference ellipsoid, km)
  2    Height (above local terrain, km)
  3    Latitude (deg.)
  4    West Longitude (deg.)
  5    Time from start (Earth seconds)
  6    Time from start (Martian Sols)
  7    Areocentric Longitude of Sun, Ls (deg.)
  8    Hour Angle for Local Time (Mars hours * 15)

Use y-code = 0 for plotable output vs x-code variable only
2 0
 For density and pressure data units, enter:
 0 for normal (MKS), 1 for log-base-10, 2 for % difference from COSPAR
2
Enter Initial Latitude (deg.), West Longitude (deg.)
22 48
Surface elevation =   -.50 km at this location
Enter Initial Height (km)
-0.5
Enter Increments in Height (km), Latitude (deg.), West Longitude (deg.),
 and Time (sec.)
10 0 0 0
Computing data.
Enter Month, Day of Month, 4-digit Year, and Max Number Positions
0 0 0 0
 Normal Termination
```

Table 4-2. Sample NAMELIST file INPUT for Batch Form of Mars-GRAM 3.34. Inline comments are appended after the ! symbol. Some FORTRAN compilers do not allow inline comments in NAMELIST data files, in which case they cannot be included in the INPUT. Some FORTRAN compilers use alternate forms of the initial and final lines of the file (e.g. &INPUT for $INPUT and / for $END is the form for Microsoft DOS FORTRAN).

```
$INPUT
  LSTFL     = 'VIKING1.LST',  ! List file name (CON for console listing)
  OUTFL     = 'VIKING1.OUT',  ! Output file name
  MONTH     = 7,          ! month of year
  MDAY      = 20,         ! day of month
  MYEAR     = 76,         ! year (4-digit; 1970-2069 can be 2-digit)
  NPOS      = 21,         ! max # positions to evaluate (0 = read data from file)
  IHR       = 12,         ! GMT hour of day
  IMIN      = 30,         ! minute of hour
  SEC       = 0.0,        ! second of minute (for initial position)
  ALS0      = 0.0,        ! starting Ls value (degrees) for dust storm (0 = none)
  INTENS    = 0.0,        ! dust storm intensity (0.0 - 3.0)
  RADMAX    = 0.0,        ! max. radius (km) of dust storm (0 or >10000 = global)
  DUSTLAT   = 0.0,        ! latitude (deg) for center of dust storm
  DUSTLON   = 0.0,        ! West longitude (deg) for center of dust storm
  F107      = 185.0,      ! 10.7 cm solar flux (10**-22 W/cm**2, at 1 AU)
  STDL      = 0.0,        ! std. dev. for thermosphere variation (-3.0 to +3.0)
  MODPERT   = 3,          ! perturbation model; 1=random, 2=wave, 3=both
  NR1       = 1001,       ! starting random number (0 < NR1 < 30000)
  NVARX     = 2,          ! x-code for plotable output (1=hgt above ref. ellipse)
  NVARY     = 0,          ! y-code for 2-D plotable output (0 for 1-D plots)
  LOGSCALE  = 0,          ! 0=regular density, 1=log(density), 2=COSPAR deviations
  FLAT      = 22.0,       ! initial latitude (N positive), degrees
  FLON      = 48.0,       ! initial longitude (West positive), degrees
  FHGT      = -0.5,       ! initial height (km), above ref. ellipse
  DELHGT    = 10.0,       ! height increment (km) between steps
  DELLAT    = 0.0,        ! latitude increment (deg) between steps
  DELLON    = 0.0,        ! West longitude increment (deg) between steps
  DELTIME   = 0.0,        ! time increment (sec) between steps
$END
```

Table 4-3. List of x-code and y-code Values and Parameters for Plotable Output (graphics) Files

```
Code               Parameter                                          SETU283
----  --------------------------------------------------              SETU284
  1    Height (above reference ellipsoid, km)                         SETU285
  2    Height (above local terrain, km)                               SETU286
  3    Latitude (deg.)                                                SETU287
  4    West Longitude (deg.)                                          SETU288
  5    Time from start (Earth seconds)                                SETU289
  6    Time from start (Martian Sols)                                 SETU290
  7    Areocentric Longitude of Sun, Ls (deg.)                        SETU291
  8    Hour Angle for Local Time (Mars hours * 15)                    SETU292

Use y-code = 0 for plotable output vs x-code variable only            SETU293
```

4.2 Program Output

Three general types of program output are: (1) a "LIST" file, containing header and descriptor information, suitable for printing or viewing by an analyst (LIST file, Table 4-4), (2) "OUTPUT" file, containing no header or descriptor information, one line per output position, suitable for reading into another program for additional analysis (OUTPUT file, in Table 4-5), and (3) set of "plotable" output files, suitable for input to a graphics program (Table 4-6).

Plot output files contain either x-y data pairs or x-y-z data triplets, determined from selected values for x-code (NVARX) and y-code (NVARY). If 1-D line-graph (x-y pair) data is the selected plot output option (y-code = 0), then the x-code variable normally appears first in the output pair. However, if the x-code variable selected for plot output is a height variable (x-code = 1 or 2) then the plot output values give the height as the second variable of the output pair. This facilitates graphs with height as the ordinate, since many graphics programs expect input data pairs in abscissa-ordinate order. If 2-D contour plot (x-y-z triplet) data is the selected plot output option (y-code ≠ 0), then the plot output variables always appear as: x-code variable, y-code variable, plot output variable. See Table 4-7 for list of plot output variables generated and associated file unit numbers (plot output files are unit numbers 21 through 28 and 30 through 34).

Table 4-7 gives the unit numbers for all program input and output files. With the exception of the screen output and the TRAJDATA input file, the program input and output files are also referred to within the program by a FILES array. The FILES array index values associated with the program input and output files are also shown in Table 4-7.

If the user desires to suppress the LIST, OUTPUT, and plotable output files (to handle output in a user-provided program), set the LIST file unit number (iup) to 0 (line BLKD 15, in BLOCK DATA routine in MARSSUBS.FOR file) and re-compile the program. The unit number associated with "screen" output (iu0), normally 6, is set to any other value by changing it at line BLKD 16 and re-compiling the program.

WRITE statements and output FORMATS used to produce the OUTPUT file are shown in the following code lines from the DATASTEP subroutine (in MARSSUBS.FOR file):

```
      If(logscale.eq.0)Then                                      DSTP346b
        WRITE(29,800)CSEC,VAR,CLAT,CLON,dens,TEMP,EWWIND,       DSTP347
     &    NSWIND,SIGD,DENSWA,ALS                                  DSTP348
      Else                                                        DSTP348a
        WRITE(29,810)CSEC,VAR,CLAT,CLON,dens,TEMP,EWWIND,       DSTP348b
     &    NSWIND,SIGD,DENSWA,ALS                                  DSTP348c
      Endif                                                       DSTP348d
    Endif                                                         DSTP348e
```

```
800 FORMAT(F10.0,3F7.2,1PE9.2,0P3F7.1,3F6.1)                    DSTP349
810 FORMAT(F10.0,3F7.2,F9.2,3F7.1,3F6.1)                        DSTP349a
```

If other variables or output formats are desired, alter appropriate program lines and re-compile the program.

The output data, some echoed from input values for a record of the options selected, are illustrated by the sample LIST file in Table 4-4. Output variables and descriptions are as follows:

MONTH, MDAY, MYEAR, IHR, IMIN, SEC, ALS0, INTENS, RADMAX, DUSTLAT, DUSTLON, F107, STDL, MODPERT, and NR1 are defined in Section 4.1.

DATE	Julian date, computed from input year, month and day
CSEC	Time (seconds) from start of simulation [also units of Mars days (sols)]
ALS	Areocentric longitude of the Sun (Ls) in degrees (angle at Mars analogous to right ascension of the Sun for the Earth)
OHGT	Height above reference ellipsoid (km)
OHGTS	Height above local terrain (km) (shown in parentheses in LIST file output)
HSCALE	Pressure scale height (km)
CLAT	Current latitude (positive North)
CLON	Current longitude (positive West)
SUNLON	Mars longitude of sub-solar point (degrees)
TLOCAL	Local solar time (in Mars hours, 1/24th sols)
TEMP	Temperature (K) at current location (also output to plot file TEMP, unit 25, Table 4-7)
PRES	Pressure (N/m^2) at current location (also output to plot file PRES, unit 26, Table 4-7; units in plot file controlled by LOGSCALE option)
DENSLO	Low (approximately -1 standard deviation) density (kg/m^3) at current location (also output to plot file DENSLO, unit 33, Table 4-7; units in plot file controlled by LOGSCALE option)

DENS Average density (kg/m^3) at current location (also output to plot file DENSAV, unit 22, Table 4-7; units in plot file controlled by LOGSCALE option)

DENSHI High (approximately +1 standard deviation) density (kg/m^3) at current location (also output to plot file DENSHI, unit 34, Table 4-7; units in plot file controlled by LOGSCALE option)

DEVLO Low density expressed as percentage deviation from COSPAR value

DEVAV Average density expressed as percentage deviation from COSPAR value

DEVHI High density expressed as percentage deviation from COSPAR value

DENSP Density perturbation (relative to mean value), expressed as percentage of mean (also output to plot file DENSRP, unit 24, Table 4-7).

EWWIND Eastward wind component (m/s) (also output to plot file EWWIND, unit 27, Table 4-7).

NSWIND Northward wind component (m/s) (also output to plot file NSWIND, unit 28, Table 4-7).

Other variables not output to LIST file but output to plot files (may be selected for output to OUTPUT file) are the following:

SIGD Standard deviation of density perturbations (percent of mean value) (output to plot file DENSRM, unit 21, Table 4-7)

DENSWA Value of the Zurek wave model density perturbation (percent of mean value) (output to plot file DENSWA, unit 23, Table 4-7)

TMAX Daily maximum temperature (K) at surface, for current location (output to plot file TMAX, unit 30, Table 4-7)

TMIN Daily minimum temperature (K) at surface, for current location (output to plot file TMIN, unit 31, Table 4-7)

TAVG Daily average temperature (K) at surface, for current location (output to plot file TAVG, unit 32, Table 4-7)

Table 4-4. LIST file (VIKING1.LST) Produced by Either Interactive or Batch Form (Page 1 of 4)

```
Mars-GRAM Version 3.34 Interactive Form - November 1, 1995
Date =  7/20/1976  Julian Date = 2442980.0  GMT Time = 12:30:  .0
F10.7 flux = 185.0 (1 AU)   68.1 (Mars),  standard deviation =   .0
Perturbation model = 3   Starting random number =  1001
Time (rel. to T0) =        .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height =   -.50 km (    .00 km)   Scale Height =  12.48 km
Latitude =   22.000 degrees       West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.     Local Time =  16.21 Mars hours
Temperature =   243.4 K           Pressure =    7.478E+02 N/m**2
Density (Low, Avg., High) =      1.502E-02   1.607E-02   1.712E-02 kg/m**3
Departure, COSPAR NH Mean =        -7.5 %      -1.0 %       5.5 %
Density perturbation =           8.76 % of mean value
Eastward Wind =     4.3 m/s       Northward Wind =     -.3 m/s
-----------------------------------------------------------------------
Time (rel. to T0) =        .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height =   9.50 km (  10.00 km)   Scale Height =  10.39 km
Latitude =   22.000 degrees       West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.     Local Time =  16.21 Mars hours
Temperature =   202.7 K           Pressure =    3.055E+02 N/m**2
Density (Low, Avg., High) =      7.619E-03   7.884E-03   8.152E-03 kg/m**3
Departure, COSPAR NH Mean =        12.9 %      16.8 %      20.8 %
Density perturbation =          -1.30 % of mean value
Eastward Wind =     1.3 m/s       Northward Wind =    -5.5 m/s
-----------------------------------------------------------------------
Time (rel. to T0) =        .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height =  19.50 km (  20.00 km)   Scale Height =   9.49 km
Latitude =   22.000 degrees       West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.     Local Time =  16.21 Mars hours
Temperature =   185.1 K           Pressure =    1.122E+02 N/m**2
Density (Low, Avg., High) =     3.038E-03   3.172E-03   3.309E-03 kg/m**3
Departure, COSPAR NH Mean =       10.2 %      15.0 %      20.0 %
Density perturbation =          -1.41 % of mean value
Eastward Wind =    -1.8 m/s       Northward Wind =    -9.0 m/s
-----------------------------------------------------------------------
Time (rel. to T0) =        .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height =  29.50 km (  30.00 km)   Scale Height =   8.72 km
Latitude =   22.000 degrees       West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.     Local Time =  16.21 Mars hours
Temperature =   170.2 K           Pressure =    3.800E+01 N/m**2
Density (Low, Avg., High) =     1.096E-03   1.168E-03   1.242E-03 kg/m**3
Departure, COSPAR NH Mean =        6.0 %      13.0 %      20.1 %
Density perturbation =          -.30 % of mean value
Eastward Wind =    -4.8 m/s       Northward Wind =   -13.9 m/s
-----------------------------------------------------------------------
Time (rel. to T0) =        .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height =  39.50 km (  40.00 km)   Scale Height =   8.19 km
Latitude =   22.000 degrees       West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.     Local Time =  16.21 Mars hours
Temperature =   159.8 K           Pressure =    1.192E+01 N/m**2
Density (Low, Avg., High) =     3.655E-04   3.903E-04   4.160E-04 kg/m**3
Departure, COSPAR NH Mean =        1.8 %       8.7 %      15.9 %
Density perturbation =          -.32 % of mean value
Eastward Wind =    -8.2 m/s       Northward Wind =   -20.1 m/s
-----------------------------------------------------------------------
Time (rel. to T0) =        .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height =  49.50 km (  50.00 km)   Scale Height =   7.67 km
Latitude =   22.000 degrees       West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.     Local Time =  16.21 Mars hours
Temperature =   149.6 K           Pressure =    3.488E+00 N/m**2
Density (Low, Avg., High) =      1.103E-04   1.220E-04   1.340E-04 kg/m**3
Departure, COSPAR NH Mean =        -4.0 %       6.3 %      16.8 %
```

Table 4-4. LIST file (VIKING1.LST) Produced by Either Interactive or Batch Form (Page 2 of 4)

```
Density perturbation =          5.36 % of mean value
Eastward Wind =   -11.9 m/s      Northward Wind =   -26.7 m/s
--------------------------------------------------------------------------
Time (rel. to T0) =       .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height =  59.50 km (  60.00 km)  Scale Height =   7.39 km
Latitude =   22.000 degrees      West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.    Local Time =  16.21 Mars hours
Temperature =   144.1 K          Pressure =    9.618E-01 N/m**2
Density (Low, Avg., High) =     3.018E-05   3.491E-05   3.973E-05 kg/m**3
Departure, COSPAR NH Mean =      -10.9 %       3.0 %      17.3 %
Density perturbation =         -9.27 % of mean value
Eastward Wind =   -16.2 m/s      Northward Wind =   -33.9 m/s
--------------------------------------------------------------------------
Time (rel. to T0) =       .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height =  69.50 km (  70.00 km)  Scale Height =   7.13 km
Latitude =   22.000 degrees      West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.    Local Time =  16.21 Mars hours
Temperature =   139.0 K          Pressure =    2.552E-01 N/m**2
Density (Low, Avg., High) =     8.298E-06   9.599E-06   1.092E-05 kg/m**3
Departure, COSPAR NH Mean =      -11.1 %       2.9 %      17.1 %
Density perturbation =         -1.71 % of mean value
Eastward Wind =   -20.7 m/s      Northward Wind =   -39.8 m/s
--------------------------------------------------------------------------
Time (rel. to T0) =       .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height =  79.50 km (  80.00 km)  Scale Height =   7.34 km
Latitude =   22.000 degrees      West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.    Local Time =  16.21 Mars hours
Temperature =   136.5 K          Pressure =    5.487E-02 N/m**2
Density (Low, Avg., High) =     1.680E-06   2.100E-06   2.549E-06 kg/m**3
Departure, COSPAR NH Mean =      -31.4 %     -14.3 %       4.1 %
Density perturbation =          3.65 % of mean value
Eastward Wind =   -23.7 m/s      Northward Wind =   -40.5 m/s
--------------------------------------------------------------------------
Time (rel. to T0) =       .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height =  89.50 km (  90.00 km)  Scale Height =   7.55 km
Latitude =   22.000 degrees      West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.    Local Time =  16.21 Mars hours
Temperature =   139.2 K          Pressure =    8.699E-03 N/m**2
Density (Low, Avg., High) =     2.227E-07   3.256E-07   4.505E-07 kg/m**3
Departure, COSPAR NH Mean =      -65.3 %     -49.3 %     -29.9 %
Density perturbation =        -31.47 % of mean value
Eastward Wind =   -22.0 m/s      Northward Wind =   -31.5 m/s
--------------------------------------------------------------------------
Time (rel. to T0) =       .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height =  99.50 km ( 100.00 km)  Scale Height =   7.90 km
Latitude =   22.000 degrees      West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.    Local Time =  16.21 Mars hours
Temperature =   144.4 K          Pressure =    1.470E-03 N/m**2
Density (Low, Avg., High) =     3.083E-08   5.284E-08   8.334E-08 kg/m**3
Departure, COSPAR NH Mean =      -81.9 %     -68.9 %     -51.0 %
Density perturbation =         -7.08 % of mean value
Eastward Wind =   -18.3 m/s      Northward Wind =   -21.2 m/s
--------------------------------------------------------------------------
Time (rel. to T0) =       .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height = 109.50 km ( 110.00 km)  Scale Height =   8.44 km
Latitude =   22.000 degrees      West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.    Local Time =  16.21 Mars hours
Temperature =   152.9 K          Pressure =    2.776E-04 N/m**2
Density (Low, Avg., High) =     4.922E-09   9.394E-09   1.618E-08 kg/m**3
Departure, COSPAR NH Mean =      -88.9 %     -78.7 %     -63.4 %
Density perturbation =        128.57 % of mean value
Eastward Wind =   -11.5 m/s      Northward Wind =    -9.8 m/s
--------------------------------------------------------------------------
```

Table 4-4. LIST file (VIKING1.LST) Produced by Either Interactive or Batch Form (Page 3 of 4)

```
Time (rel. to T0) =        .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height = 119.50 km ( 120.00 km)   Scale Height =  11.18 km
Latitude =   22.000 degrees       West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.     Local Time =  16.21 Mars hours
Temperature =   199.8 K           Pressure =    7.618E-05 N/m**2
Density (Low, Avg., High) =      1.317E-09   1.959E-09   2.886E-09 kg/m**3
Departure, COSPAR NH Mean =        -89.6 %     -84.5 %     -77.2 %
Density perturbation =          -17.49 % of mean value
Eastward Wind =     -4.4 m/s      Northward Wind =    -3.2 m/s
-----------------------------------------------------------------------
Time (rel. to T0) =        .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height = 129.50 km ( 130.00 km)   Scale Height =  14.53 km
Latitude =   22.000 degrees       West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.     Local Time =  16.21 Mars hours
Temperature =   255.1 K           Pressure =    3.512E-05 N/m**2
Density (Low, Avg., High) =      4.979E-10   6.990E-10   9.288E-10 kg/m**3
Departure, COSPAR NH Mean =        -87.5 %     -82.4 %     -76.6 %
Density perturbation =           1.15 % of mean value
Eastward Wind =     -2.0 m/s      Northward Wind =    -2.8 m/s
-----------------------------------------------------------------------
Time (rel. to T0) =        .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height = 139.50 km ( 140.00 km)   Scale Height =  16.60 km
Latitude =   22.000 degrees       West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.     Local Time =  16.21 Mars hours
Temperature =   285.8 K           Pressure =    1.851E-05 N/m**2
Density (Low, Avg., High) =      2.262E-10   3.241E-10   4.298E-10 kg/m**3
Departure, COSPAR NH Mean =        -80.3 %     -71.8 %     -62.6 %
Density perturbation =           1.73 % of mean value
Eastward Wind =      -.8 m/s      Northward Wind =    -1.7 m/s
-----------------------------------------------------------------------
Time (rel. to T0) =        .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height = 149.50 km ( 150.00 km)   Scale Height =  18.01 km
Latitude =   22.000 degrees       West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.     Local Time =  16.21 Mars hours
Temperature =   302.9 K           Pressure =    1.039E-05 N/m**2
Density (Low, Avg., High) =      1.124E-10   1.687E-10   2.300E-10 kg/m**3
Departure, COSPAR NH Mean =        -77.1 %     -65.7 %     -53.2 %
Density perturbation =          37.30 % of mean value
Eastward Wind =      -.4 m/s      Northward Wind =    -1.0 m/s
-----------------------------------------------------------------------
Time (rel. to T0) =        .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height = 159.50 km ( 160.00 km)   Scale Height =  19.10 km
Latitude =   22.000 degrees       West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.     Local Time =  16.21 Mars hours
Temperature =   312.5 K           Pressure =    6.064E-06 N/m**2
Density (Low, Avg., High) =      5.869E-11   9.339E-11   1.324E-10 kg/m**3
Departure, COSPAR NH Mean =        -76.6 %     -62.8 %     -47.2 %
Density perturbation =          -35.75 % of mean value
Eastward Wind =      -.3 m/s      Northward Wind =     -.5 m/s
-----------------------------------------------------------------------
Time (rel. to T0) =        .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height = 169.50 km ( 170.00 km)   Scale Height =  20.07 km
Latitude =   22.000 degrees       West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.     Local Time =  16.21 Mars hours
Temperature =   317.9 K           Pressure =    3.638E-06 N/m**2
Density (Low, Avg., High) =      3.170E-11   5.362E-11   7.946E-11 kg/m**3
Departure, COSPAR NH Mean =        -77.2 %     -61.4 %     -42.8 %
Density perturbation =          -8.43 % of mean value
Eastward Wind =      -.2 m/s      Northward Wind =     -.3 m/s
-----------------------------------------------------------------------
Time (rel. to T0) =        .0 sec. (    .000 sols)    Ls =   97.0 deg.
Height = 179.50 km ( 180.00 km)   Scale Height =  21.06 km
Latitude =   22.000 degrees       West Longitude =   48.000 degrees
```

Table 4-4. LIST file (VIKING1.LST) Produced by Either Interactive or Batch Form (Page 4 of 4)

```
Sun Longitude =  111.188 deg.      Local Time =  16.21 Mars hours
Temperature =   321.0 K            Pressure =    2.236E-06 N/m**2
Density (Low, Avg., High) =       1.757E-11   3.158E-11   4.897E-11 kg/m**3
Departure, COSPAR NH Mean =        -78.3 %     -61.1 %     -39.6 %
Density perturbation =            -.26 % of mean value
Eastward Wind =     -.2 m/s        Northward Wind =     -.1 m/s
-------------------------------------------------------------------------
Time (rel. to T0) =        .0 sec. (     .000 sols)    Ls =   97.0 deg.
Height = 189.50 km ( 190.00 km)   Scale Height =  22.15 km
Latitude =   22.000 degrees        West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.      Local Time =  16.21 Mars hours
Temperature =   322.7 K            Pressure =    1.407E-06 N/m**2
Density (Low, Avg., High) =       9.968E-12   1.899E-11   3.078E-11 kg/m**3
Departure, COSPAR NH Mean =        -79.5 %     -60.9 %     -36.7 %
Density perturbation =           -3.07 % of mean value
Eastward Wind =     -.1 m/s        Northward Wind =     -.1 m/s
-------------------------------------------------------------------------
Time (rel. to T0) =        .0 sec. (     .000 sols)    Ls =   97.0 deg.
Height = 199.50 km ( 200.00 km)   Scale Height =  23.43 km
Latitude =   22.000 degrees        West Longitude =   48.000 degrees
Sun Longitude =  111.188 deg.      Local Time =  16.21 Mars hours
Temperature =   323.7 K            Pressure =    9.066E-07 N/m**2
Density (Low, Avg., High) =       5.795E-12   1.164E-11   1.967E-11 kg/m**3
Departure, COSPAR NH Mean =        -80.6 %     -61.1 %     -34.3 %
Density perturbation =           -4.85 % of mean value
Eastward Wind =     -.1 m/s        Northward Wind =      .0 m/s
-------------------------------------------------------------------------
```

Table 4-5. OUTPUT file (VIKING1.OUT) Produced by Either Interactive or Batch Form (Heading added for readability)

Time	Hgt	Lat	Lon	Dens %COSPAR	Temp	EWind	Nwind	Dev	Wave	Ls
0.	.00	22.00	48.00	-1.00	243.4	4.3	-.3	.9	5.6	97.0
0.	10.00	22.00	48.00	16.80	202.7	1.3	-5.5	1.9	1.5	97.0
0.	20.00	22.00	48.00	15.05	185.1	-1.8	-9.0	2.9	1.4	97.0
0.	30.00	22.00	48.00	13.01	170.2	-4.8	-13.9	3.9	2.4	97.0
0.	40.00	22.00	48.00	8.71	159.8	-8.2	-20.1	4.8	1.6	97.0
0.	50.00	22.00	48.00	6.27	149.6	-11.9	-26.7	4.9	4.9	97.0
0.	60.00	22.00	48.00	3.04	144.1	-16.2	-33.9	4.9	8.8	97.0
0.	70.00	22.00	48.00	2.89	139.0	-20.7	-39.8	4.9	8.8	97.0
0.	80.00	22.00	48.00	-14.28	136.5	-23.7	-40.5	11.9	8.8	97.0
0.	90.00	22.00	48.00	-49.33	139.2	-22.0	-31.5	26.2	8.8	97.0
0.	100.00	22.00	48.00	-68.94	144.4	-18.3	-21.2	40.9	8.8	97.0
0.	110.00	22.00	48.00	-78.75	152.9	-11.5	-9.8	51.1	8.8	97.0
0.	120.00	22.00	48.00	-84.54	199.8	-4.4	-3.2	31.2	8.8	97.0
0.	130.00	22.00	48.00	-82.41	255.1	-2.0	-2.8	22.0	8.8	97.0
0.	140.00	22.00	48.00	-71.78	285.8	-.8	-1.7	22.6	8.8	97.0
0.	150.00	22.00	48.00	-65.68	302.9	-.4	-1.0	26.1	8.8	97.0
0.	160.00	22.00	48.00	-62.78	312.5	-.3	-.5	30.7	8.8	97.0
0.	170.00	22.00	48.00	-61.41	317.9	-.2	-.3	35.7	8.8	97.0
0.	180.00	22.00	48.00	-61.06	321.0	-.2	-.1	40.9	8.8	97.0
0.	190.00	22.00	48.00	-60.94	322.7	-.1	-.1	46.0	8.8	97.0
0.	200.00	22.00	48.00	-61.10	323.7	-.1	.0	50.8	8.8	97.0

Table 4-6. Sample Plotable Output File (DENSAV) [first variable = average density (% COSPAR difference), second variable = height]

-.9976	.0000
16.80	10.00
15.05	20.00
13.01	30.00
8.714	40.00
6.274	50.00
3.039	60.00
2.891	70.00
-14.28	80.00
-49.33	90.00
-68.94	100.0
-78.75	110.0
-84.54	120.0
-82.41	130.0
-71.78	140.0
-65.68	150.0
-62.78	160.0
-61.41	170.0
-61.06	180.0
-60.94	190.0
-61.10	200.0

Table 4-7. File Names Used in the Mars-GRAM Programs

FILES Array Index	File Name (description of file)	Unit Number
1	LSTFL (default name = LIST)	iup (default=11 BLKD 15) Set to 0 to suppress LIST and graphics file output
2	DENSRM (std. deviation in density)	21
3	DENSAV (average density)	22
4	DENSWA (wave model perturbations)	23
5	DENSRP (random perturbations)	24
6	TEMP (temperatures)	25
7	PRES (pressure)	26
8	EWWIND (eastward wind component)	27
9	NSWIND (northward wind component)	28
10	OUTFL (default name = OUTPUT)	29
11	HEIGHTS.DAT (terrain heights)	9
12	TMAX (daily max. surf. temp.)	30
13	TMIN (daily min. surf. temp.)	31
14	TAVG (daily avg. surf. temp.)	32
15	DENSLO (low density, -1 sigma)	33
16	DENSHI (high density, +1 sigma)	34
17	COSPAR.DAT (COSPAR data arrays)	10
N/A	TRAJDATA (trajectory input)	7
N/A	Unnamed screen output	iu0 (default=6 BLKD 16)

4.3 How to Use Mars-GRAM Batch Form as Subroutines in Other Programs

Stripped of all embedded comment statements, the batch form program (MARSGRMB.FOR) consists of the following lines of code:

```
      DOUBLE PRECISION DATE0
      Real NSWIND
      Integer EOF
      Call Setup(CHGT,CLAT,CLON,CSEC,DATE0,RHO,DELHGT,DELLAT,DELLON,
     & DELTIME,MAXNUM)
      DO 900 I = 0,MAXNUM
        Call Datastep(I,CHGT,CLAT,CLON,CSEC,DATE0,RHO,EOF,DELHGT,
     &   DELLAT,DELLON,DELTIME,TEMP,PRES,DENSLO,DENS,DENSHI,DENSP,
     &   EWWIND,NSWIND)
        If (EOF .eq. 1)Goto 999
  900 Continue
  999 STOP ' Normal Termination'
      END
```

All variables are initialized through SETUP subroutine which must be called once at beginning of program. Atmospheric parameters are evaluated at successive locations (either trajectory or automatic profile mode) with DATASTEP subroutine. Any desired output (or transfer to other subroutines) is done within the loop that calls DATASTEP (i.e. just after "Call Datastep" statement). To suppress the normal LIST, OUTPUT, and graphics file output, set iup=0 at line BLKD 15 in the BLOCK DATA routine (in the MARSSUBS.FOR file).

To embed SETUP and DATASTEP subroutines into a program that computes trajectory or orbit positions, the calling program keeps current position updated (height CHGT, latitude CLAT, longitude CLON, at time after start of the run CSEC) and computes velocity components from which displacements in height, latitude, and longitude can be computed for any desired time step.

A program using SETUP and DATASTEP as subroutines takes the following form (added pseudo-code is shown in brackets):

```
      DOUBLE PRECISION DATE0
      Real NSWIND
      Integer EOF
      Call Setup(CHGT,CLAT,CLON,CSEC,DATE0,RHO,DELHGT,DELLAT,DELLON,
     & DELTIME,MAXNUM)

        [Comment: Evaluate the atmospheric parameters at the initial
         position by calling Datastep with I=0, so that the current
         position is not updated.]

        Call Datastep(0,CHGT,CLAT,CLON,CSEC,DATE0,RHO,EOF,DELHGT,
     &   DELLAT,DELLON,DELTIME,TEMP,PRES,DENSLO,DENS,DENSHI,DENSP,
     &   EWWIND,NSWIND)

 100   Continue

        [Use the vertical velocity in the trajectory to compute the
         height displacement, DELHGT, the north-south velocity to
         compute the latitude displacement, DELLAT, and the east-west
         velocity to compute the longitude displacement, DELLON, based
         on the desired time displacement, DELTIME. These displacement
         values will override the values initially defined in the
         SETUP routine]


        Call Datastep(1,CHGT,CLAT,CLON,CSEC,DATE0,RHO,EOF,DELHGT,
```

```
&    DELLAT,DELLON,DELTIME,TEMP,PRES,DENSLO,DENS,DENSHI,DENSP,
&    EWWIND,NSWIND)

     [Comment:  With I ≠ 0 in the call to Datastep, the position
     is updated by the position displacements, before the new
     atmospheric parameters are evaluated.]

     [Write out the current position (CHGT, CLAT, CLON), as it
      has been updated by Datastep, along with any other output
      desired.  Any other desired analysis of the variables at the
      current position can also be done here in the program.]

     [If the last position and time has been computed, go to 999.
      Otherwise, go to 100.]

999 STOP ' Normal Termination'
    END
```

5. DIAGNOSTIC AND PROGRESS MESSAGES

5.1 Interactive Form Main Program Mars-GRAM

The interactive form Mars-GRAM main program prompts the user for input options. Some prompt statements provide legal values that can be acceptable for the prompted input option. If the program repeats the prompt message after entered input, a value outside the legal range was input. Corrective action is to input a value within the legal range. Prompt message ceases and operation continues.

Other specific diagnostic or progress messages follow. All STOP messages and messages written to unit iu0 normally appear on the interactive user screen (default value for iu0 is 6, set at line BLKD 16 in BLOCK DATA).

Code Producing the Message:

```
50      Write(iu0,60)files(j),ierr(j)                                   MARS108
60      Format(1x,a12,' File open error! Error =',i5)                   MARS109
```

Purpose:

To indicate an error occurred in attempt to open one of the 17 files in FILES array (see Table 4-7 for file names). Message gives file name and error code that caused the (first encountered) problem.

Remedy:

Depending on meaning of error code on compiler system, take appropriate corrective action to ensure proper file open operation. Rerun program.

Code Producing the Message:

```
65 Stop ' Error or EOF on COSPAR.DAT file!'                             MARS112g
```

Purpose:

To indicate an End-of-File or read error occurred on COSPAR.DAT file (file may not exist or path to this file may not be set appropriately).

Remedy:

Correct error condition (i.e., copy or restore file to right location). Rerun program.

Code Producing the Message:

```
80 If (lat .ne. -100+10*i)Stop ' Error reading HEIGHTS.DAT!'         MARS116
```

Purpose:

To indicate a latitude read from HEIGHTS.DAT file was an unexpected value (file is missing, corrupted, or sized wrong based on parameters nlat and nlon).

Remedy:

Correct problem with HEIGHTS.DAT file. Rerun program.

Code Producing the Message:

```
260   Write(iu0)270                                                    MARS144
270   format(' Unable to open Trajectory Data file!')                  MARS145
```

Purpose:

To indicate an error occurred in attempt to open trajectory input data file TRAJDATA (file may not exist or may have wrong path set to its location).

Remedy:

Correct problem with TRAJDATA file. Rerun program.

Code Producing the Message:

```
      If (als0 .gt. 0.)Write(iu0,*)' Ls outside range. ',          MARS177
   &    ' No dust storm assumed.'                                  MARS178
```

Purpose:

To indicate user attempted to simulate a dust storm for a value of areocentric longitude of the Sun (Ls) outside range 180 to 320. Message reminds user that dust storms cannot be simulated for Ls values outside this range and no dust storm is being simulated during the run.

Remedy:

None necessarily required. If a dust storm case is desired, select a different Ls (180 < Ls < 320). Rerun program.

Code Producing the Message:

```
      Write(iu0,*)' Intensity must be between 0 and 3'          MARS187
```

Purpose:

To indiciate a dust storm intensity < 0.0 or > 3.0 was entered.

Remedy:

Input a dust storm intensity between 0.0 and 3.0. Rerun program.

Code Producing the Message:

```
      Write(iu0,*)' F10.7 must be between 50 and 450'            MARS213
```

Purpose:

To indicate an F10.7 solar flux <50 or >450 was entered.

Remedy:

Input an F10.7 value between 50 and 450. Rerun program.

Code Producing the Message:

```
      Write(iu0,*)' Std. deviations must be between -3 and +3'         MARS217
```

Purpose:

To indicate a standard deviation in the Stewart thermosphere model < -3 or > +3 was entered (nominal value is 0.0).

Remedy:

Input a standard deviation value between -3 and +3. Rerun program.

Code Producing the Message:

```
  430 Write(iu0,440)                                                     MARS312
  440 FORMAT(' Computing data.')                                         MARS313
```

Purpose:

To indicate all data are input successfully and computations underway.

Remedy:

Complete the run. Time required depends on size of trajectory input file or number of automatic profile positions to be computed.

Code Producing the Message:

```
9998 Stop ' Error termination on opening or reading input file'        MARS323
```

Purpose:

To indicate a termination message if file open error occurred on trajectory file or other files in FILES array.

Remedy:

Correct file problem, according to preceding error message. Rerun program.

Code Producing the Message:

```
9999 STOP ' Normal Termination'                                        MARS324
```

Purpose:

To indicate successful completion of program run.

Remedy:

None required.

5.2 Batch Form Main Program MARSGRMB

The batch form Marsgrmb main program does not produce diagnostic or progress messages other than the normal termination message for output to the STOP messages location.

Code Producing the Message:

```
999 STOP ' Normal Termination'                                          MARB117
```

Purpose:

To indicate successful completion of program run.

Remedy:

None required.

5.3 Subroutine SETUP

Several diagnostic or progress messages are produced by the SETUP subroutine. SETUP is called by the batch form only (not the interactive form) of the program.

Code Producing the Message:

```
50      Write(iu0,60)files(j),ierr(j)                                SETU141
60      Format(1x,a12,' File open error! Error =',i5)                SETU142
```

Purpose:

To indicate an error occurred in attempt to open one of the 17 files in FILES array (see separate Table 4-7 for file names). Message gives file name and error code that caused the (first encountered) problem.

Remedy:

Depending on meaning of error code on compiler system, take appropriate corrective action to ensure proper file open operation. Rerun program.

Code Producing the Message:

```
65 Stop ' Error or EOF on COSPAR.DAT file!'                          SETU146g
```

Purpose:

To indicate an End-of-File or read error occurred on COSPAR.DAT file (file may not exist or path to this file may not be set appropriately).

Remedy:

Correct error condition (i.e., copy or restore file to right location). Rerun program.

Code Producing the Message:

```
80 If (lat .ne. -100+10*i)Stop ' Error reading HEIGHTS.DAT!'        SETU150
```

Purpose:

To indicate a latitude read from HEIGHTS.DAT file was an unexpected value (file is missing, corrupted, or sized wrong based on parameters nlat and nlon).

Remedy:

Correct problem with HEIGHTS.DAT file. Rerun program.

Code Producing the Message:

```
     Write(iu0,91)                                                  SETU157
91   Format(' Input error in month, day or year.')                  SETU158
```

Purpose:

To indicate an illegal value of month, day, or year input via the NAMELIST INPUT file.

Remedy:

Correct erroneous value(s). Rerun program.

Code Producing the Message:

```
     Write(iu0,92)                                                  SETU170
92   Format(' Input error in hour, minute or seconds.')             SETU171
```

Purpose:

To indicate an illegal value of hour, minute, or second input via the NAMELIST INPUT file.

Remedy:

Correct erroneous value(s). Rerun program.

Code Producing the Message:

```
260   Write(iu0,270)                                         SETU191
270   format(' Unable to open Trajectory Data file!')        SETU192
```

Purpose:

To indicate an error occurred in attempt to open trajectory input data file TRAJDATA (file may not exist or may have wrong path set to its location).

Remedy:

Correct problem with TRAJDATA file. Rerun program.

Code Producing the Message:

```
      If (als0 .gt. 0.)Write(iu0,*)' Ls outside range. ',    SETU216
     &  ' No dust storm assumed.'                            SETU217
```

Purpose:

To indicate user attempted to simulate a dust storm for a value of areocentric longitude of the Sun (Ls) outside range 180 to 320. Message reminds user that dust storms cannot be simulated for Ls values outside this range and that no dust storm is being simulated during the run.

Remedy:

None necessarily required. If a dust storm case is desired, select a different Ls (180 < Ls < 320). Rerun program.

Code Producing the Message:

```
        Write(iu0,*)' Intensity must be between 0 and 3'     SETU222
```

Purpose:

To indicate a dust storm intensity < 0.0 or > 3.0 was entered.

Remedy:

Input a dust storm intensity between 0.0 and 3.0 . Rerun program.

Code Producing the Message:

```
      Write(iu0,*)' F10.7 must be between 50 and 450'              SETU233
```

Purpose:

To indicate an F10.7 solar flux <50 or >450 was entered.

Remedy:

Input an F10.7 value between 50 and 450. Rerun program.

Code Producing the Message:

```
      Write(iu0,*)' Std. deviations must be between -3 and +3'     SETU237
```

Purpose:

To indicate a standard deviation in the Stewart thermosphere model <-3 or >+3 was entered (nominal value is 0.0).

Remedy:

Input a standard deviation value between -3 and +3. Rerun program.

Code Producing the Message:

```
      Write(iu0,291)                                               SETU241
  291 Format(' Error in perturbation model number.')               SETU242
```

Purpose:

To indicate a value of the perturbation model code less than 1 or greater than 3 was input via NAMELIST INPUT file.

Remedy:

Input a legal model number less than 1 or greater than 3. Rerun program.

Code Producing the Message:

```
      Write(iu0,292)                                            SETU247
 292  Format(' Error in starting random number.')               SETU248
```

Purpose:

To indicate a random number seed value of less than 1 or greater than 29999 was input via NAMELIST INPUT file.

Remedy:

Input a random number seed value greater than 0 and less than 30000. Rerun program.

Code Producing the Message:

```
      Write(iu0,381)                                            SETU295
 381  Format(' x-code or y-code input error.')                  SETU296
```

Purpose:

To indicate a value of x-code outside the range 1 to 8, or y-code outside range 0 to 8 was input via NAMELIST INPUT file.

Remedy:

Select x-code and y-code values for plotable output, according to the following codes:

```
     & ' Code              Parameter'/                                SETU283
     & ' ----   -------------------------------------------------'/  SETU284
     & '   1    Height (above reference ellipsoid, km)'/              SETU285
     & '   2    Height (above local terrain, km)'/                    SETU286
     & '   3    Latitude (deg.)'/                                     SETU287
     & '   4    West Longitude (deg.)'/                               SETU288
     & '   5    Time from start (Earth seconds)'/                     SETU289
     & '   6    Time from start (Martian Sols)'/                      SETU290
     & '   7    Areocentric Longitude of Sun, Ls (deg.)'/             SETU291
     & '   8    Hour Angle for Local Time (Mars hours * 15)'//        SETU292
     & ' Use y-code = 0 for plotable output vs x-code variable only') SETU293
```

Rerun program.

Code Producing the Message:

```
      Write(iu0,382)                                              SETU307
 382  Format(' Error in first latitude or longitude.')            SETU308
```

Purpose:

To indicate a starting latitude absolute value greater than 90, or a starting longitude less than 0 or greater than 360, was input via NAMELIST input file.

Remedy:

Input a legal value of starting latitude and longitude. Rerun program.

Code Producing the Message:

```
9998 Stop ' Error termination! Check the LIST file for messages.'    SETU324
```

Purpose:

To indicate termination, i.e., a fatal error condition occurred. Normally error conditions are described by message preceding this on screen output. For some options and error conditions, additional error information may appear in LIST output file.

Remedy:

Correct the error(s) noted. Rerun program.

5.4 Subroutines in the File MARSSUBS

The subroutines in the MARSSUBS file produce diagnostic messages when certain error conditions occur. As an option, various diagnostic outputs are triggered by the user from within the Stewart thermosphere model, by setting FLAG (an integer) to a non-zero.

Code Producing the Message:

```
      IF (IFAULT .EQ. 1)STOP ' PPND ERROR'                            DSTP 61
```

Purpose:

To indicate a fatal error occurred in random number routine PPND. This is abnormal and may indicate a system-level problem.

Remedy:

Rerun the program with a different random number seed (NR1). If problem persists, run system level diagnostic tests to see if RANDOM and PPND perform properly.

Code Producing the Message:

```
 9998 Stop ' Error termination reading data file!'                    DSTP353
```

Purpose:

To indicate an error occurred while reading trajectory file TRAJDATA.

Remedy:

Correct data in TRAJDATA file. Rerun program.

Code Producing the Message:

```
C      if (FLAG .GT. 0)Write(iu0,170)LSUN,DFAO,DZ                     DZDS 23
C  170 FORMAT(' FROM PROC. DZDUST-- LSUN,DFAO,DZ = ',3F8.2)          DZDS 24
```

Purpose:

This code is currently "commented out" (inactivated). If comment characters are removed from column 1 and if FLAG is set to a non-zero value and passed as an argument into subroutine DZDUST, then this optional diagnostic output is produced.

Remedy:

After tests are completed and diagnostic output is not desired, set FLAG to 0 (in Stewart2 subroutine, line STW2 20).

Code Producing the Message:

```
      if(FLAG.gt.0)Write(iu0,*)' RREF, RAU, GZ = ',RREF,RAU,GZ          STW2 32

      if(FLAG.gt.0)Write(iu0,*)' ZF,CHGT,ZZF = ',ZF,CHGT,ZZF            STW2 44

        Write(iu0,*)' RAU,FBARR,RREF,DR,DUST = '                        STW2 49
        Write(iu0,150)RAU,FBARR,RREF,DR,DUST                            STW2 50
  150   FORMAT(F6.2,F7.2,F8.2,2F6.2,F8.2)                               STW2 51
        Write(iu0,*)' '                                                 STW2 52
        Write(iu0,*)'FROM PROC. DRAG-- ZF,RF,TINF,TF = '                STW2 53
        Write(iu0,160)ZF,RF,TINF,TF                                     STW2 54
  160   FORMAT(F7.2,F8.2,3F7.2)                                         STW2 55
```

Purpose:

To produce this optional diagnostic output, set value of FLAG to a non-zero value.

Remedy:

After tests are completed and diagnostic output is not desired, set FLAG to 0 (in Stewart2 subroutine, line STW2 20).

Code Producing the Message:

```
      if (FLAG .GT. 0) then                                             THRM135
        Write(iu0,220)ZZF,TZ,MOLWTG,TOTALPRZ,TOTALNDZ,(NDZ(K),K=0,3),   THRM136
     &   TOTALMDZ,(NDZ(K),K=4,8)                                        THRM137
  220   FORMAT(2F6.1,F5.1,6E10.3/17X,6E10.3)                            THRM138
```

Purpose:

To produce this optional diagnostic output, set the value of FLAG to a non-zero value and pass as an argument into subroutine THERMOS.

Remedy:

After tests are completed and diagnostic output is not desired, set FLAG to 0 (in Stewart2 subroutine, line STW2 20).

Page
5-16
Blank

6. FUTURE PLANS

The NASA Marshall Space Flight Center (MSFC) and NASA Jet Propulsion Laboratory (JPL) are working a joint program of future upgrades and applications development for Mars-GRAM. The following is a brief outline of the development features under consideration.

6.1 Improved Mars Thermosphere Model

The Stewart model thermosphere that serves as the basis for the upper altitudes in Mars-GRAM was designed to give global average values of density and other variables as a function of height and time. This means that latitude-longitude variations of thermospheric density, temperature, and pressure are unrealistic in Mars-GRAM. Since the wind components are based on extremely small horizontal pressure gradients (through the areostrophic or thermal wind relation), the winds estimated by Mars-GRAM in the thermosphere region are unrealistically small.

To improve the Mars-GRAM thermospheric values, output data from the Bougher et al. (1988) Mars Thermospheric Global Circulation Model (MTGCM) will be used to derive appropriate modifier factors for latitude and time-of-day (longitude), for application to the Stewart model values. Comparisons with MTGCM results may also improve the Mars-GRAM (Stewart) model dependence on the effects of solar activity (10.7 cm solar flux). With comparison of wind output from MTGCM, the wind model in Mars-GRAM, including the viscous modification model, also has potential for improvement.

6.2 Climatic Changes Since the Mariner-Viking (1970's) Time Period

Because Mars-GRAM was developed from parameterizations to atmospheric data observed by the Mariner and Viking missions, the model is representative of the Mars atmosphere during the 1970's. Recent observations by Clancy et al. (1990) indicate, in response to atmospheric cooling as the dusty atmosphere has cleared in the last two decades, that current Mars temperature profiles are distinctly cooler than those observed in the Viking era. Comparisons of Mars-GRAM mid-latitude average temperature profiles with recent data from Clancy (provided by Rich Zurek, private communication), indicate about 20 K cooling at the 40-50 km height range, about 15 K at 20-30 km, but little change at 5-10 km. This temperature change could have significant effects on atmospheric density at high altitudes and, as Clancy et al. (1990) pointed out, is of considerable importance in planning for Mars missions that involve aerobraking.

Consider Mars-GRAM profiles of temperature, density and pressure to be $T_0(z)$, $\rho_0(z)$, and $p_0(z)$, respectively, with the perfect gas law and hydrostatic relations applicable, namely,

$$p_0 = \rho_0 R T_0 \qquad , \tag{23}$$

and

$$dp_0 / dz = - \rho_0 g = - p_0(z) / H(z) \qquad . \tag{24}$$

Consider also that climatic changes up to the present time have induced new profiles of temperature, density and pressure, $T_1(z)$, $\rho_1(z)$ and $p_1(z)$. Application of equations (23) and (24) to both the original and the new atmospheric states yields the relation for density:

$$\delta\rho / \rho_0 = (\rho_1 - \rho_0) / \rho_0 = (T_0 / T_1) \exp\left(\int_0^z [(T_1 - T_0) / (T_0 H)] \right) dz' - 1 \ . \tag{25}$$

Consider the simple example in which temperature has uniformly decreased by 5% between the surface and 100 km, i.e., $\delta T / T_0 = (T_1 - T_0) / T_0 \approx -0.05$ at all altitudes. Since the scale height, H, is about 10 km, the integral term in equation (25) would have a value of about -0.5 and the density change at 100 km altitude from equation (25) would be $\delta\rho / \rho_0 = -36\%$. In more general cases, the exact details of the density effect at altitude would, of course, depend on the profile of the temperature changes $(T_1 - T_0)$ and the profile of the scale height values H(z).

Another Mars-GRAM enhancement under consideration by MSFC and JPL is to use new atmospheric temperature profiles from the Clancy data (and/or or from prior missions, after the first new Mars mission) to derive a climate-shift option (i.e. to estimate current conditions with the climatic shift option on, or to estimate Viking era conditions with the new climate shift option off).

Mars-GRAM as an Operational Tool for Aerobraking

MSFC and JPL are also interested in developing operational tools, based on Mars-GRAM, to use during the aerobraking phase of future Mars missions. Two potential approaches are to:

(1) Use Mars-GRAM densities evaluated near the periapsis position and time to compare with densities derived from the "real-time" orbit/trajectory analysis. Keep a time-series history of a "figure of merit" to monitor the trends of necessary adjustments to convert Mars-GRAM derived densities to actual densities for the aerobraking period. The figure of merit may be simply a ratio of Mars-GRAM value to the real time value or, perhaps, the value of the standard deviation (parameter STDL in the Stewart thermosphere model) required to get Mars-GRAM output to agree with real-time derived density. Once the observed trend in the figure of merit is established, use, in conjunction with Mars-GRAM calculations, to make short-term forecasts of the atmospheric density to be encountered over the next several passes through periapsis.

(2) Use other "real-time" measurements of temperature profiles (e.g., Clancy-type ground-based microwave observations or temperature profiler data from sensors onboard the Mars orbiter) to derive estimates of necessary corrections to be applied to Mars-GRAM output. In this mode, estimates of density deviations from Mars-GRAM are computed via equation (25), using estimates of observed temperature profile deviations from Mars-GRAM profiles.

Page
6-4
Blank

7. BIBLIOGRAPHY

Bougher, S.W., et al. (1988): "Mars Thermospheric General Circulation Model: Calculations for the Arrival of Phobos at Mars." Geophys. Res. Let., 15(13).

Clancy, R.T., D.O Muhleman and G.L. Berge (1990): "Global Changes in the 0-70 km Thermal Structure of the Mars Atmosphere Derived from 1975 to 1989 Microwave CO Spectra." J. Geophys. Res., 95(B9).

Davies, D.W. (1979): "Effects of Dust on the Heating of Mars' Surface and Atmosphere." J. Geophys. Res., 84(B14).

James, Bonnie F. and C. G. Justus (1993): "The Mars Global Reference Atmosphere Model (Mars-GRAM) Release #2." NASA MSFC Electromagnetics and Environments Branch Technical Report 3-1-93 (Appendix C).

Johnson, D.L. and Bonnie F. James (grant monitors), C.G. Justus and George Chimonas (1989): "The Mars Global Reference Atmospheric Model (Mars-GRAM)." Final report under NASA Grant No. NAG8-078, Georgia Tech Project G-35-685, October 8, 1989 (Appendix B).

Justus, C.G. (1990): "A Mars Global Reference Atmospheric Model (Mars-GRAM) for Mission Planning and Analysis." AIAA 90-004, 28th Aerospace Sciences Meeting, Reno, NV, January.

Justus, C.G. (1991): "Mars Global Reference Atmospheric Model for Mission Planning and Analysis." J. Spacecraft and Rockets, 28(2).

Justus, C.G. and M.V. Paris (1985): "Modeling Solar Spectral Irradiance and Radiance at the Bottom and Top of a Cloudless Atmosphere." J. Climate Appl. Meteorol., 24(3.

Pitts, David E. et al. (1990?): "The Mars Atmosphere: Observations and Model Profiles for Mars Missions." NASA JSC-24455.

Stewart, A.I.F. (1987): Revised Time Dependent Model of the Martian Atmosphere for use in Orbit Lifetime and Sustenance Studies." Final report JPL PO# NQ-802429, March 26, 52 pp.

Page
7-2
Blank

APPENDIX A

DESCRIPTION OF THE MARS-GRAM PROGRAM AND SUBROUTINES

This Appendix gives descriptions of the interactive form (Mars-GRAM) main program, the batch form (MARSGRMB) main program, the subroutine SETUP (used only by the batch form), and each of the functions and subroutines in the file MARSSUBS (in alphabetical order by the line number code for the MARSSUBS routines). References to functions, subroutines, and variables used appear as either all capital letters or as upper and lower case. The FORTRAN compiler is assumed to be case-insensitive.

Each description gives: (1) function or subroutine name, (2) four-character line number code, (3) brief description of purpose of subroutine, (4) subroutine(s) that call function or subroutine, (5) common blocks used, and (6) list and descriptions of input variables, output variables, and local variables (those not passed through common blocks). For a list of variables in common blocks, see Table 2-3.

A technical discussion of methods used by each function or subroutine is also given. Reference is made to the actual line of program code, including line numbers, for easy cross-reference to the program code.

1. The Mars-GRAM Main Program (Interactive Form)

Main Program: Mars-GRAM (version 3.34) **Code:** MARS

Description: Interactive form of the Mars-GRAM program.

Called By: N/A

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

COSPARNH, DATACOM, FILENAME, RANDCOM, TERHGT, THERM, WAVEDAT

Input Variables (Read from interactive input):

Name	Type	Description
ALS0	REAL*4	time (Ls value) for beginning of dust storm
DC	REAL*4	COSPAR density array
DELHGT	REAL*4	height increment (km)
DELLAT	REAL*4	northward latitude increment (degrees)
DELLON	REAL*4	westward longitude increment (degrees)
DELTIME	REAL*4	time increment (seconds)
DUSTLAT	REAL*4	latitude of local-scale dust storm
DUSTLON	REAL*4	longitude of local-scale dust storm
F107	REAL*4	10.7 cm solar flux at Earth position (1AU)
FHGT	REAL*4	first height (km)
FLAT	REAL*4	first latitude (degrees)
FLON	REAL*4	first longitude (degrees West)
IHR	INTEGER*4	hour of day (UTC or GMT)
IMIN	INTEGER*4	minute of hour
INTENS	REAL*4	intensity level for dust storm (0.0 to 3.0)
LAT	INTEGER*4	latitude for reading HEIGHTS.DAT input
LOGSCALE	INTEGER*4	output units for pressure and density (0=MKS, 1 = log-base-10 MKS, 2=%deviation from COSPAR)
LSTFL	CHAR*12	Name of LIST file
MDAY	INTEGER*4	day of month
MODPERT	INTEGER*4	perturbation model (1=random, 2=wave, 3=both)
MONTH	INTEGER*4	month of year
MYEAR	INTEGER*4	year (4 digit; can be 2-digit if 1970 to 2069)
NPOS	INTEGER*4	maximum number of automatically-generated positions
NR1	INTEGER*4	random number seed (1-30,000)
NVARX	INTEGER*4	x-coordinate plot variable (See MARS262-MARS274)
NVARY	INTEGER*4	y-coordinate plot variable (See MARS262-MARS274)
OUTFL	CHAR*12	Name of OUTPUT file
PC	REAL*4	COSPAR pressure array
RADMAX	REAL*4	maximum radius of dust storm (km; global if radmax=0 or radmax>10,000)
SEC	REAL*4	second of minute
STDL	REAL*4	number of standard deviations for thermospheric variation (-3.0 to +3.0)
TC	REAL*4	COSPAR temperature array
TH	REAL*4	terrain height array, read from HEIGHTS.DAT
ZC	REAL*4	COSPAR height array

Output Variables (Includes echo out of some input variables):

Name	Type	Description
ALS	REAL*4	(Ls) areocentric longitude of sun
ALS0	REAL*4	time (Ls value) for beginning of dust storm
DATE	REAL*8	Julian date
DUSTLAT	REAL*4	latitude of local-scale dust storm
DUSTLON	REAL*4	longitude of local-scale dust storm
F107	REAL*4	10.7 cm solar flux at Earth position (1AU); program also writes out value at Mars position
IHR	INTEGER*4	hour of day (UTC or GMT)
IMIN	INTEGER*4	minute of hour
INTENS	REAL*4	intensity level for dust storm
MDAY	INTEGER*4	day of month
MODPERT	INTEGER*4	perturbation model (1=random, 2=wave, 3=both)
MONTH	INTEGER*4	month of year
MYEAR	INTEGER*4	year (4 digit; can be 2-digit if 1970 to 2069)
NPOS	INTEGER*4	maximum number of automatically-generated positions
NR1	INTEGER*4	random number seed
RADMAX	REAL*4	maximum radius of dust storm
SEC	REAL*4	second of minute
STDL	REAL*4	number of standard deviations for thermospheric variation
THGT	REAL*4	local elevation of surface (km)

Local Variables (not passed through commons):

Name	Type	Description
ALS	REAL*4	(Ls) areocentric longitude of the sun
CHGT	REAL*4	current height
CLAT	REAL*4	current latitude
CLON	REAL*4	current longitude
CSEC	REAL*4	current time from start (seconds)
DATE	REAL*8	Julian date
DATE0	REAL*8	initial Julian data
DELHGT	REAL*4	height increment
DELLAT	REAL*4	latitude increment
DELLON	REAL*4	longitude increment
DELTIME	REAL*4	time increment
DENS	REAL*4	atmospheric density
DENSHI	REAL*4	high (approx. + 1 sigma) density
DENSLO	REAL*4	low (approx. - 1 sigma) density
DENSP	REAL*4	perturbation in density (% of mean)
EOF	INTEGER*4	End-of-file flag for trajectory file
EWWIND	REAL*4	eastward vector wind component (m/s)
FHGT	REAL*4	first height (km)
FILES	CHAR*12	array of file names (see list of values elsewhere)
FLAT	REAL*4	first latitude
FLON	REAL*4	first longitude
GZERO	REAL*4	gravity acceleration at z=0
I	INTEGER*4	index variable
IDAY	INTEGER*4	day-of-year for beginning of each month
IERR	INTEGER*4	file open error flag array (equivalent to IERR1-IERR17)
IERR1	INTEGER*4	file open error flag
IERR10	INTEGER*4	file open error flag
IERR11	INTEGER*4	file open error flag
IERR12	INTEGER*4	file open error flag
IERR13	INTEGER*4	file open error flag
IERR14	INTEGER*4	file open error flag

IERR15	INTEGER*4	file open error flag
IERR16	INTEGER*4	file open error flag
IERR17	INTEGER*4	file open error flag
IERR2	INTEGER*4	file open error flag
IERR3	INTEGER*4	file open error flag
IERR4	INTEGER*4	file open error flag
IERR5	INTEGER*4	file open error flag
IERR6	INTEGER*4	file open error flag
IERR7	INTEGER*4	file open error flag
IERR8	INTEGER*4	file open error flag
IERR9	INTEGER*4	file open error flag
IHR	INTEGER*4	hour of day
IMIN	INTEGER*4	minute of hour
J	INTEGER*4	index value
L	INTEGER*4	index value
LAT	INTEGER*4	latitude for reading terrain heights
MARSAU	REAL*4	orbital position of Mars (in AU)
MAXNUM	INTEGER*4	maximum number of positions to compute
MDAY	INTEGER*4	day of month
MONTH	INTEGER*4	month of year
MYEAR	INTEGER*4	year
NDAY	INTEGER*4	day of year
NLAT	INTEGER*4	number of latitudes in terrain height array
NLON	INTEGER*4	number of longitudes in terrain height array
NR1	INTEGER*4	random number seed
NSWIND	REAL*4	northward vector wind component
PRES	REAL*4	atmospheric pressure
RHO	REAL*4	random number for sequence of perturbations
SEC	REAL*4	seconds of minute
SUNLAT	REAL*4	latitude of sub-solar position on surface
SUNLON	REAL*4	longitude of sub-solar position on surface
TEMP	REAL*4	atmospheric temperature (K)
THGT	REAL*4	height of terrain surface above reference ellipsoid
XYEAR	REAL*8	intermediate year for Julian day calculation

Methodology:

The program prompts the user for values of the necessary input. Depending on the options selected, some input parameters assume default values and it is not necessary to input a value. Several input variables are also echoed back as output variable on the LIST file. All subroutines required by Mars-GRAM interactive form are contained in the marssubs file.

Mars-GRAM opens all input and output files, interactively reads the necessary input values and steps through the output positions, generating and writing the output. If the trajectory option is selected (NPOS=0), positions are read from the TRAJDATA file, until and end-of-file is encountered. If an automatically-generated profile of positions is used (NPOS>0), output continues until the maximum allowed number of output positions (MAXNUM=NPOS-1) has been exceeded.

If the time is within the period for dust storm formation (180 < Ls <320, Ls = areocentric longitude of sun), then a local-scale or a global-scale dust storm can be simulated. Position and or size of the dust storm are controlled by input values of DUSTLAT, DUSTLON, RADMAX and INTENS.

2. The Mars-GRAM Main Program (Batch Form)

Main Program: Marsgrmb (version 3.34) **Code:** MARB

Description: Batch form of the Mars-GRAM program.

Called By: N/A

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables:

There are no input variables directly to Marsgrmb. Input data are read, in NAMELIST form, by the subroutine SETUP. An example of the NAMELIST input is as follows:

```
$INPUT
 LSTFL    = 'LIST',   ! List file name (CON for console listing)
 OUTFL    = 'OUTPUT',! Output file name
 MONTH    = 7,        ! month of year
 MDAY     = 20,       ! day of month
 MYEAR    = 76,       ! year (4-digit; 1970-2069 can be 2-digit)
 NPOS     = 21,       ! max # positions to evaluate (0 = read data
                      !                           from TRAJDATA file)
 IHR      = 12,       ! GMT hour of day
 IMIN     = 30,       ! minute of hour
 SEC      = 0.0,      ! second of minute (for initial position)
 ALS0     = 0.0,      ! starting Ls value (degrees) for dust storm
                      !                                   (0 = none)
 INTENS   = 0.0,      ! dust storm intensity (0.0 - 3.0)
 RADMAX   = 0.0,      ! max. radius (km) of dust storm (0 or
                      !                             >10000 = global)
 DUSTLAT  = 0.0,      ! latitude (deg) for center of dust storm
 DUSTLON  = 0.0,      ! West longitude (deg) of center of dust storm
 F107     = 185.0,    ! 10.7 cm solar flux (10**-22 W/cm**2 at 1 AU)
 STDL     = 0.0,      ! std. dev. for thermosphere variation (-3.0
                      !                                     to +3.0)
 MODPERT  = 3,        ! perturbation model; 1=random, 2=wave, 3=both
 NR1      = 1001,     ! starting random number (0 < NR1 < 30000)
 NVARX    = 2,        ! x-code for plotable output (1=hgt above ref.
                      !                                     ellipse)
 NVARY    = 0,        ! y-code for 2-D plotable output (0 for 1-D
                      !                                       plots)
 LOGSCALE = 0,        ! 1 for log-base-10 scale plots, 0 for linear
                      !   scale, 2 for % deviation from COSPAR
 FLAT     = 22.0,     ! initial latitude (N positive), degrees
 FLON     = 48.0,     ! initial longitude (West positive), degrees
 FHGT     = -0.5,     ! initial height (km), above ref. ellipse
 DELHGT   = 10.0,     ! height increment (km) between steps
 DELLAT   = 0.0,      ! latitude increment (deg) between steps
 DELLON   = 0.0,      ! West longitude increment (deg) between steps
 DELTIME  = 0.0,      ! time increment (sec) between steps
$END
```

Output Variables:

There are no output variables that are produced directly by Marsgrmb. Standard LIST and OUTPUT files (as well as the output files for graphics input, units 21-34) may be produced by subroutine DATASTEP. The Marsgrmb batch form is designed so that the output variables from DATASTEP may easily be output by the user (or passed to other user subroutines), in whatever format is desired. Output variables from DATASTEP are:

```
      TEMP = temperature (K)                                        MARB 98
      PRES = pressure (N/m**2)                                      MARB 99
      DENSLO = nominal low density (kg/m**3), approx. -1 sigma      MARB100
      DENS   = mean density (kg/m**3)                               MARB101
      DENSHI = nominal high density (kg/m**3), approx. +1 sigma     MARB102
      DENSP = density perturbation about mean (% of mean)           MARB103
      EWWIND = eastward wind component (m/s)                        MARB104
      NSWIND = northward wind component (m/s)                       MARB105
```

If none of the LIST file or OUTPUT file or graphics file output is desired, this is accomplished by setting file unit parameter iup = 0, in line BLKD 16.

Local Variables (not passed through commons):

Name	Type	Description
CHGT	REAL*4	current height
CLAT	REAL*4	current latitude
CLON	REAL*4	current longitude
CSEC	REAL*4	current time from start (seconds)
DATE0	REAL*8	initial Julian date
DELHGT	REAL*4	height increment
DELLAT	REAL*4	latitude increment
DELLON	REAL*4	longitude increment
DELTIME	REAL*4	time increment
DENS	REAL*4	atmospheric density
DENSHI	REAL*4	high (approx. + 1 sigma) density
DENSLO	REAL*4	low (approx - 1 sigma) density
DENSP	REAL*4	perturbation in density (% of mean)
EOF	INTEGER*4	End-of-file flag for trajectory file
EWWIND	REAL*4	eastward vector wind component
I	INTEGER*4	index variable
MAXNUM	INTEGER*4	maximum number of positions to compute
NSWIND	REAL*4	northward vector wind component
PRES	REAL*4	atmospheric pressure
RHO	REAL*4	random number for sequence of perturbations
TEMP	REAL*4	atmospheric temperature (K)

Methodology:

Stripped of all of the embedded comment statements, the batch form Marsgrmb consists of only the following lines of code. All variables are initialized through the SETUP subroutine and parameters are evaluated at successive locations (either trajectory or automatic profile mode) with the DATASTEP subroutine. Any desired output (or transfer to other subroutines) can be done within the loop that calls DATASTEP.

```
      DOUBLE PRECISION DATE0
      Real NSWIND
      Integer EOF
      Call Setup(CHGT,CLAT,CLON,CSEC,DATE0,RHO,DELHGT,DELLAT,DELLON,
     & DELTIME,MAXNUM)
      DO 900 I = 0,MAXNUM
        Call Datastep(I,CHGT,CLAT,CLON,CSEC,DATE0,RHO,EOF,DELHGT,
     &   DELLAT,DELLON,DELTIME,TEMP,PRES,DENSLO,DENS,DENSHI,DENSP,
     &   EWWIND,NSWIND)
```

```
      If (EOF .eq. 1)Goto 999
900 Continue
999 STOP ' Normal Termination'
    END
```

3. Description of the SETUP Subroutine (for the Batch Form)

Subroutine: SETUP **Code:** SETU

Description: Opens files and reads input data for the Marsgrmb batch form program.

Called By: Marsgrmb (main)

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

COSPARNH, DATACOM, FILENAME, RANDCOM, TERHGT, THERM

Input Variables:

Input data are read, in NAMELIST form, by the subroutine SETUP. An example of the NAMELIST input is as follows:

```
$INPUT                                                          MARB 18
 LSTFL    = 'LIST',   ! List file name (CON for console listing)   MARB 19
 OUTFL    = 'OUTPUT',! Output file name                            MARB 20
 MONTH    = 7,        ! month of year                              MARB 21
 MDAY     = 20,       ! day of month                               MARB 22
 MYEAR    = 76,       ! year (4-digit; 1970-2069 can be 2-digit)   MARB 23
 NPOS     = 11,       ! max # positions to evaluate (0 = read data MARB 24
                      !                     from TRAJDATA file)    MARB 25
 IHR      = 12,       ! GMT hour of day                            MARB 26
 IMIN     = 30,       ! minute of hour                             MARB 27
 SEC      = 0.0,      ! second of minute (for initial position)    MARB 28
 ALS0     = 0.0,      ! starting Ls value (degrees) for dust storm MARB 29
                      !                               (0 = none)   MARB 30
 INTENS   = 0.0,      ! dust storm intensity (0.0 - 3.0)           MARB 31
 RADMAX   = 0.0,      ! max. radius (km) of dust storm (0 or       MARB 32
                      !                         >10000 = global)   MARB 33
 DUSTLAT  = 0.0,      ! latitude (deg) for center of dust storm    MARB 34
 DUSTLON  = 0.0,      ! West longitude (deg) of center of dust storm MARB 35
 F107     = 185.0,    ! 10.7 cm solar flux (10**-22 W/cm**2 at 1 AU) MARB 36
 STDL     = 0.0,      ! std. dev. for thermosphere variation (-3.0 MARB 37
                      !                                 to +3.0)   MARB 38
 MODPERT  = 3,        ! perturbation model; 1=random, 2=wave, 3=both MARB 39
 NR1      = 1001,     ! starting random number (0 < NR1 < 30000)   MARB 40
 NVARX    = 1,        ! x-code for plotable output (1=hgt above ref. MARB 41
                      !                                 ellipse)   MARB 42
 NVARY    = 0,        ! y-code for 2-D plotable output (0 for 1-D  MARB 43
                      !                                   plots)   MARB 44
 LOGSCALE = 0,        ! 1 for log-base-10 scale plots, 0 for linear MARB 45
                      !   scale, 2 for % deviation from COSPAR     MARB 46
 FLAT     = 22.0,     ! initial latitude (N positive), degrees     MARB 47
 FLON     = 48.0,     ! initial longitude (West positive), degrees MARB 48
 FHGT     = -0.5,     ! initial height (km), above ref. ellipse    MARB 49
 DELHGT   = 10.0,     ! height increment (km) between steps        MARB 50
 DELLAT   = 0.0,      ! latitude increment (deg) between steps     MARB 51
 DELLON   = 0.0,      ! West longitude increment (deg) between steps MARB 52
 DELTIME  = 0.0,      ! time increment (sec) between steps         MARB 53
$END
```

Output Variables (not passed through commons):

Name	Type	Description
CHGT	REAL*4	current height
CLAT	REAL*4	current latitude
CLON	REAL*4	current longitude
CSEC	REAL*4	current time (seconds)
DATE0	REAL*4	initial Julian date
DHGT	REAL*4	height increment
DLAT	REAL*4	latitude increment
DLON	REAL*4	longitude increment
DTIME	REAL*4	time increment (seconds)
MAXNUM	INTEGER*4	maximum number of positions to compute
RHO	REAL*4	random number for sequence of perturbations

Local Variables (not passed through commons):

Name	Type	Description
ALS	REAL*4	areocentric longitude of the sun (Ls)
DATE	REAL*8	Julian date
DELHGT	REAL*4	height increment
DELLAT	REAL*4	latitude increment
DELLON	REAL*4	longitude increment
DELTIME	REAL*4	time increment
FILES	CHAR*12	file name array (see Table with list of files)
GZERO	REAL*4	acceleration of gravity at z = 0
I	INTEGER*4	index variable
IDAY	INTEGER*4	day-of-year for beginning of each month
IERR	INTEGER*4	file open error flag array (equivalent to IERR1-IERR17)
IERR1	INTEGER*4	file open error flag
IERR10	INTEGER*4	file open error flag
IERR11	INTEGER*4	file open error flag
IERR12	INTEGER*4	file open error flag
IERR13	INTEGER*4	file open error flag
IERR14	INTEGER*4	file open error flag
IERR15	INTEGER*4	file open error flag
IERR16	INTEGER*4	file open error flag
IERR17	INTEGER*4	file open error flag
IERR2	INTEGER*4	file open error flag
IERR3	INTEGER*4	file open error flag
IERR4	INTEGER*4	file open error flag
IERR5	INTEGER*4	file open error flag
IERR6	INTEGER*4	file open error flag
IERR7	INTEGER*4	file open error flag
IERR8	INTEGER*4	file open error flag
IERR9	INTEGER*4	file open error flag
J	INTEGER*4	index variable
L	INTEGER*4	index variable
LAT	INTEGER*4	latitude for reading terrain heights
MARSAU	REAL*4	orbital position of Mars (in AU)
NDAY	INTEGER*4	day of month
NLAT	INTEGER*4	number of latitudes in terrain height array
NLON	INTEGER*4	number of longitudes in terrain height array
SUNLAT	REAL*4	latitude of sub-solar position on surface
SUNLON	REAL*4	longitude of sub-solar position on surface
THGT	REAL*4	height of terrain surface above reference ellipsoid
XYEAR	REAL*8	intermediate year for Julian day calculation

Methodology:

Reads NAMELIST input values (see above), computes Julian day, open output files and passes data to the Marsgrmb main program through the common blocks.

4. Description of the MARSSUBS.FOR Functions and Subroutines

Function: Alb **Code:** ALBL

Description: Surface albedo as a function of areocentric longitude of the sun (Ls) and latitude.

Called By: Tsurface

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
als . . .	REAL*4	areocentric longitude of sun (Ls), degrees
alat. . .	REAL*4	latitude, degrees

Output Variables (not passed through commons):

Name	Type	Description
Alb . . .	REAL*4	surface albedo

Local Variables (not passed through commons):

Name	Type	Description
al. . . .	REAL*4	intermediate variable for albedo calculation
alsc. . .	REAL*4	Ls phase for polar hood cloud
caprad. .	REAL*4	polar cap radius, degrees
cldrad. .	REAL*4	polar hood cloud radius, degrees
pi180 . .	REAL*4	factor for degrees to radians
radc. . .	REAL*4	radius amplitude for polar hood cloud

Methodology:

Computes albedo of non-polar surface as

```
      al = 0.32 - 0.12*cos(pi180*alat)                          ALBL  9
```

Computes radius for polar hood cloud from

```
      cldrad = radc*(1. + cos(pi180*(als - alsc)))              ALBL 22
```

where alsc = 280., radc = 20. for North polar hood cloud and alsc = 60. and radc = 27. for South polar hood cloud. If the latitude is poleward of the polar hood cloud radius, the surface albedo is increased by 15 %

```
      If (abs(alat) .ge. 90. - cldrad)al = 1.15*al              ALBL 24
```

The radius of the polar cap is computed from the polecap function

```
caprad = polecap(alat,als)                                    ALBL 26
```

If the latitude is poleward of the polar cap radius, the surface albedo is set to 0.6

```
If (abs(alat) .ge. 90. - caprad)al = 0.6                      ALBL 28
```

Function: ampint **Code:** AMPN

Description: Interpolate wave amplitudes, linearly on input variable

Called By: Wavepert

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
amp1. . .	REAL*4	amplitude at 1st value of input variable (x1)
amp2. . .	REAL*4	amplitude at 2nd value of input variable (x2)
dx. . . .	REAL*4	relative difference in input [(x-x1)/(x2-x1)]

Output Variables (not passed through commons):

Name	Type	Description
ampint. .	REAL*4	interpolated value of wave amplitude

Local Variables (not passed through commons):

Name	Type	Description
None		

Methodology:

Interpolates linearly to position x, between amp1 (at x1) and amp2 (at x2) by the relation

$$ampint = amp1 + (amp2 - amp1)*dx \qquad \text{AMPN 2}$$

where dx = (x - x1)/(x2-x1). The input variable x may be height or latitude.

Subroutine: ATMOS2 **Code:** ATM2

Description: Driver routine to call either the STEWART2 thermosphere model or the stratos, lower altitude model, depending on the current height

Called By: Datastep

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

THERM

Input Variables (not passed through commons):

Name	Type	Description
ALS. . . .	REAL*4	areocentric longitude of the sun (Ls), degrees
ALS0 . . .	REAL*4	initial value of areocentric longitude of the sun
CHGT . . .	REAL*4	current height (km)
CLAT . . .	REAL*4	current latitude (degrees)
CLON . . .	REAL*4	current West longitude (degrees)
DUSTA. . .	REAL*4	dust storm magnitude (0-1) for daily amplitude
DUSTM. . .	REAL*4	dust storm magnitude (0-1) for average T and p
INTENS . .	REAL*4	dust storm intensity (0.0 - 3.0)
IU0. . . .	INTEGER*4	unit number for screen output messages
MARSAU . .	REAL*4	Mars orbital radius (AU)
SUNLAT . .	REAL*4	latitude of sub-solar position (degrees)
SUNLON . .	REAL*4	West longitude of sub-solar position (degrees)
Z0	REAL*4	local terrain height rel. to ref. ellipsoid (km)

Output Variables (not passed through commons):

Name	Type	Description
BRUNTF . .	REAL*4	Brunt-Vaisala frequency
DENST. . .	REAL*4	atmospheric density (kg/m^3) at output position
DENSURF. .	REAL*4	atmospheric density (kg/m^3) at surface
H.	REAL*4	scale height (km) at output position
LWFCTR . .	REAL*4	lower deviation factor for density
PRES . . .	REAL*4	atmospheric pressure (N/m^2)
RSC. . . .	REAL*4	areocentric radius to output position (km)
TAVG . . .	REAL*4	daily average surface temperature (K)
TEMP . . .	REAL*4	temperature at output position (K)
TFACTOR. .	REAL*4	density perturbation factor at base of thermosphere
TMAX . . .	REAL*4	daily maximum surface temperature (K)
TMIN . . .	REAL*4	daily minimum surface temperature (K)
UPFCTR . .	REAL*4	upper deviation factor for density
ZF	REAL*4	height of base of thermosphere (km)

Local Variables (not passed through commons):

Name	Type	Description
AMF	REAL*4	molecular weight at base of thermosphere
AMHI. . . .	REAL*4	molecular weight for high density
AMLO. . . .	REAL*4	molecular weight for low density
AMW	REAL*4	average molecular weight, surface to turbopause
BVF	REAL*4	Brunt-Vaisala frequency
BVF0 . . .	REAL*4	Brunt-Vaisala frequency at surface
DENSHI. . .	REAL*4	high (approx. +1 sigma) density
DENSLO. . .	REAL*4	low (approx. -1 sigma) density

DLON. . . .	REAL*4	longitude difference between sun and output position
DR.	REAL*4	Correction to ZF for pressure variation
DUST. . . .	REAL*4	Correction to ZF for dust storm
ES.	REAL*4	long-term standard deviations about nominal ZF
GAM	REAL*4	temperature lapse rates between significant heights
GAM0. . . .	REAL*4	mean lapse rates, for use in perturbation model
GOR	REAL*4	gravity divided by gas constant
GPH	REAL*4	areopotential heights at significant levels
GPHGT . . .	REAL*4	areopotential height difference for interpolation
GZ.	REAL*4	gravity at output position
GZERO . . .	REAL*4	surface gravity
HHI	REAL*4	scale height for high density
HLO	REAL*4	scale height for low density
I	INTEGER*4	index variable
J	INTEGER*4	index variable
P	REAL*4	array of pressures at significant levels
PAVG. . . .	REAL*4	daily average surface pressure
PF.	REAL*4	intermediate value in pressure calculation
PFAC. . . .	REAL*4	Seasonal relative pressure variation
PRESHI. . .	REAL*4	pressure for high density
PRESLO. . .	REAL*4	pressure for low density
PSURF . . .	REAL*4	local position surface pressure
RBAR. . . .	REAL*4	gas constant for Martian atmosphere up to turbopause
RREF. . . .	REAL*4	reference radius to Cain 6.1 mb surface
RSTAR . . .	REAL*4	Universal gas constant
SHGT. . . .	REAL*4	height for call to thermosphere model
SMA	REAL*4	semi-major axis of Mars orbit
T	REAL*4	array of temperatures at significant levels
T0BAR . . .	REAL*4	average surface temperature without dust storm effect
TEMPHI. . .	REAL*4	temperature for high density
TEMPLO. . .	REAL*4	temperature for low density
TF.	REAL*4	intermediate value in temperature calculation
TGRAD . . .	REAL*4	temperature gradient used in Brunt-Vaisala calculation
TIME. . . .	REAL*4	local solar time
TO.	REAL*4	factor used in correction to ZF for pressure variation
TSURF . . .	REAL*4	average surface temperature wit dust storm effect
Z	REAL*4	areographic heights at significant levels

Methodology:

Evaluates surface temperature (K), by calling Tsurface (ATM2 56). Evaluates acceleration of gravity at altitude CHGT by calling RELLIPS (ATM2 59). Gets mean temperature lapse rates, for use in perturbation model, by calling Gamma (ATM2 62).

Evaluates temperatures at significant levels (0.,5.,15.,30.,50.,75. areopotential km) by calling Temps (ATM2 69). Evaluates surface pressure, N/m^2, by calling Psurface (ATM2 71). Evaluates pressure at significant levels by calling Pressure (ATM2 74). Gets surface density from perfect gas law

```
densurf = psurf/(Rbar*Tsurf)                                    ATM2 76
```

Computes Brunt Vaisala frequency at surface

```
bvf0 = gzero*(tgrad + gzero/Cp(Tsurf))/Tsurf                    ATM2 79
if (bvf0 .lt. 0.0)bvf0 = 1.0E-5                                 ATM2 80
bvf0 = Sqrt(bvf0)                                               ATM2 81
```

If the current height is below -5 km, the surface conditions are returned and the subroutine is exited (ATM2 82 - ATM2 93). If the current height is below z(5) (areographic height for 75 km areopotential height), ATMOS2 uses the lapse rates from the significant levels to interpolate for temperature, pressure and density

```
      TEMP = T(j) - gam(j+1)*gphgt                                    ATM2101
      If (abs(gam(j+1)) .gt. 0.001)then                               ATM2102
        PRES = p(j)*exp(goR*alog(TEMP/T(j))/gam(j+1))                 ATM2103
      Else                                                            ATM2104
        PRES = p(j)*exp(-goR*gphgt/T(j))                              ATM2105
      Endif                                                           ATM2106
      DENST = PRES/(Rbar*TEMP)                                        ATM2107
```

and solves for the Brunt-Vaisala frequency and high and low density factors

```
      bvf = gz*(tgrad + gz/Cp(TEMP))/TEMP                             ATM2112
      if (bvf .le. 0.0)bvf = 1.0E-5                                   ATM2113
      Bruntf = Sqrt(bvf)                                              ATM2114
C...  UPFCTR = density and temperature part of un-damped mountain      ATM2115
C     waves                                                           ATM2116
      UPFCTR = (bvf0*Bruntf/gz)*Sqrt(densurf/DENST)                   ATM2117
      LWFCTR = 0.0                                                    ATM2118
```

If the current height is above z(5), then conditions are evaluated at height ZF, the base of the thermosphere (ATM2143 and ATM2146). If the current height is above z(5) but below ZF, values are found by interpolation, using the Stratos subroutine (ATM2150 - ATM2163). If the current height is above the base of the thermosphere, values are computed from the STEWART2 thermosphere model (ATM2164 - ATM2172).

Subroutine: Block Data **Code:** BLKD

Description: Loads data values into common blocks, for use by various subroutines in the program

Called By: N/A

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

DATACOM, WAVEDAT

Input Variables (not passed through commons):

Name	Type	Description
None		

Output Variables (not passed through commons):

Outputs the variables in the commons DATACOM and WAVEDAT

Local Variables (not passed through commons):

Name	Type	Description
None		

Methodology:

Uses data statements to fill the arrays and assign values to the variables in the commons DATACOM and WAVEDAT. These commons are used by several subroutines (see Table 2-2).

Subroutine: cospar **Code:** COSP

Description: Computes COSPAR northern hemisphere mean values of pressure, density and temperature as a function of height.

Called By: Datastep

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

cosparnh

Input Variables (not passed through commons):

Name	Type	Description
z.	REAL*4	height (km) at which to evaluate COSPAR data

Input arrays of COSPAR temperature (tc, K), pressure (pc, mb) and density (dc, g/m**3) come from the common COSPARNH, from data in the COSPAR.DAT file.

Output Variables (not passed through commons):

Name	Type	Description
p.	REAL*4	COSPAR pressure (N/m^2) at height z
rho. . . .	REAL*4	COSPAR density (kg/m^3) at height z
t.	REAL*4	COSPAR temperature (K) at height z

Local Variables (not passed through commons):

Name	Type	Description
aexp . . .	REAL*4	exponent for constant lapse rate interpolation
dz	REAL*4	relative height displacement for interpolation
H.	REAL*4	scale height for isothermal interpolation
iz	INTEGER*4	height index for interpolation
R.	REAL*4	interpolated gas law constant
R1	REAL*4	gas constant at lower interpolation height
R2	REAL*4	gas constant at upper interpolation height

Methodology:

Uses an array of COSPAR data values (loaded in from the COSPAR.DAT file). The height interval is 1 km from -5 to 130 km and 10 km from 130 to 360 km. Computed height index, iz, is for the COSPAR height value just below the desired height z (COSP 10 - COSP 14). Return zero values if the height is outside the range -5 to 360 km. (COSP 16 - COSP 21).

Uses linear interpolation on temperature

```
dz = (z - zc(iz))/(zc(iz+1) - zc(iz))                         COSP 24
t = tc(iz) + (tc(iz+1) - tc(iz))*dz                           COSP 25
```

Uses a constant lapse rate, power-law interpolation (unless the layer is isothermal)

```
      aexp = Alog(pc(iz+1)/pc(iz))/Alog(tc(iz+1)/tc(iz))            COSP 28
      p = 100.*pc(iz)*(t/tc(iz))**aexp                              COSP 29
```

Uses an exponential (constant scale height) interpolation if isothermal conditions prevail

```
      H = (zc(iz+1) - zc(iz))/Alog(pc(iz)/pc(iz+1))                 COSP 31
      p = 100.*pc(iz)*exp(-(z-zc(iz))/H)                            COSP 32
```

Computes density by linearly interpolating on gas constant and applying the perfect gas law relation

```
      R1 = pc(iz)/(dc(iz)*tc(iz))                                   COSP 35
      R2 = pc(iz+1)/(dc(iz+1)*tc(iz+1))                             COSP 36
      R = R1 + (R2-R1)*dz                                           COSP 37
C     density from perfect gas law (and convert units to kg/m**3)   COSP 38
      rho = 10.*p/(R*t)                                             COSP 39
```

Function: Cp **Code:** CPOT

Description: Specific heat at constant pressure, for a CO2 atmosphere, as a function of temperature.

Called By: ATMOS2

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
T	REAL*4	temperature (K)

Output Variables (not passed through commons):

Name	Type	Description
Cp. . . .	REAL*4	specific heat (J kg^{-1} K^{-1})

Local Variables (not passed through commons):

Name	Type	Description
None		

Methodology:

Computes Cp(T) from formula

```
Cp = 639.5 + 0.123687*T + 0.00200225*T*T                    CPOT   4
```

Subroutine: Dustfact **Code:** DSTF

Description: Computes relative dust storm intensity factors dustM and dustA as a function of the time since start of the storm, (als - als0), measured in Ls angle (degrees), and as a function of the storm intensity, intens. dustM is for relative magnitude of effect on daily average temperature and pressure. dustA is for relative magnitude of effect on diurnal (and semi-diurnal) amplitudes for temperature and pressure.

Called By: Datastep

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

DATACOM

Input Variables (not passed through commons):

Name	Type	Description
ALS	REAL*4	Areocentric longitude of sun (Ls)
CLAT. . . .	REAL*4	current latitude
CLON. . . .	REAL*4	current longitude
HGT	REAL*4	current height

Output Variables (not passed through commons):

Name	Type	Description
DUSTA . . .	REAL*4	relative magnitude for dust storm effects on daily variation amplitudes for temperature and pressure
DUSTM . . .	REAL*4	relative magnitude for dust storm effects on daily average temperature and pressure

Local Variables (not passed through commons):

Name	Type	Description
DEW	REAL*4	east-west component of distance from dust storm center
DLS	REAL*4	Ls difference from starting Ls (Ls0) for dust storm
DNS	REAL*4	north-south distance component from dust storm center
HGTDUST . .	REAL*4	vertical size to top of dust storm
RAD	REAL*4	radial distance from dust storm center
RADDUST . .	REAL*4	full horizontal radius of dust storm size
SIZEFACT. .	REAL*4	shape factor to compute DUSTA and DUSTM at location

Methodology:

Computes Ls difference (dls, degrees) from Ls value at start of dust storm (Ls0), and returns zero dust storm effect if dls < 0 or dls > 48.

```
dls = als - als0
If (dls .le. 0.0 .or. dls .gt. 48.0)then
  dustM = 0.0
  dustA = 0.0
  Return
Endif
```

Computes an initial factor value dustM that increases linearly from 0 to 1 as dls increases from 0 to 6 degrees, decreases from 1 back to 0 as dls varies between 24 and 48 degrees, or dustM = 1 otherwise.

```
      If (dls .le. 6.0)then
        dustM = dls/6.
      Else If (dls .ge. 24.)then
        dustM = 2. - dls/24.
      Else
        dustM = 1.0
      Endif
```

Computes an initial factor value dustA that increases linearly from 0 to 1 as dls increases from 0 to 9, decreases from 1 back to 0 as dls varies between 18 and 48 degrees, or dustA = 1 otherwise.

```
      If (dls .le. 9.0)then
        dustA = dls/9.0
      Else If (dls .ge. 18.)then
        dustA = (48. - dls)/30.
      Else
        dustA = 1.0
      Endif
```

Computes the size factor, sizefact, based on the position from the center of the dust storm and the maximum horizontal and vertical size of the storm.

```
      sizefact = 1.0
      If (radmax .ne. 0.0)Then
        sizefact = 0.0
        dns = DTR*Rref*(CLAT - dustlat)
        dew = DTR*Rref*cos(DTR*CLAT)*(CLON - dustlon)
        rad = Sqrt(dns**2 + dew**2)
        raddust = dustM*radmax
        hgtdust = dustM*dusthgt
        If (rad .lt. 2.0*raddust .and. HGT .lt. 2.0*hgtdust)
     &   sizefact = 0.25*(1.0 + cos(90.*DTR*rad/raddust))*
     &   (1.0 + cos(90.*DTR*HGT/hgtdust))
      Endif
```

Computes the final dust storm factor values dustM and dustA, including the effects of time from start of the storm, the size factor based on position within the storm, and the storm intensity (0-3).

```
      dustM = sizefact*dustM*intens/3.
      dustA = sizefact*dustA*intens/3.
```

Subroutine: Datastep **Code:** DSTP

Description: Driver routine for calling the atmospheric models and evaluating the winds and the density perturbations.

Called By: MarsGRAM (Main), Marsgrmb (Main)

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

COSPARNH, DATACOM, FILENAME

Input Variables (not passed through commons):

Name	Type	Description
CHGT. . . .	REAL*4	current height
CLAT. . . .	REAL*4	current latitude
CLON. . . .	REAL*4	current West longitude
CSEC. . . .	REAL*4	current time
DATE0 . . .	REAL*4	Julian date for initial position
DELHGT. . .	REAL*4	height displacement between successive positions
DELLAT. . .	REAL*4	latitude displacement between successive positions
DELLON. . .	REAL*4	longitude displacement between successive positions
DELTIME . .	REAL*4	time displacement between successive positions
EOF	INTEGER*4	flag for end of file on trajectory input
I	INTEGER*4	position counter index
RHO	REAL*4	previous density perturbation value for perturbation model

Output Variables (not passed through commons):

Name	Type	Description
DENS. . . .	REAL*4	average atmospheric density
DENSHI. . .	REAL*4	high density (approx. +1 sigma)
DENSLO. . .	REAL*4	low density (approx. -1 sigma)
DENSP . . .	REAL*4	density perturbation (% from mean value)
EWWIND. . .	REAL*4	Eastward wind component
NSWIND. . .	REAL*4	Northward wind component
PRES. . . .	REAL*4	atmospheric pressure
TEMP. . . .	REAL*4	atmospheric temperature

Local Variables (not passed through commons):

Name	Type	Description
ABSWLAT . .	REAL*4	absolute value of latitude, for wind calculations
ALOGDENS. .	REAL*4	log (base 10) density
ALS	REAL*4	areocentric longitude of the sun (Ls)
AMP1. . . .	REAL*4	diurnal amplitude in Zurek wave model
AMP2. . . .	REAL*4	semi-diurnal amplitude in Zurek wave model
AREAHGT . .	REAL*4	area-averaged terrain height
BRUNTF. . .	REAL*4	Brunt-Vaisala frequency
BVF	REAL*4	dummy argument in ATMOS2 for Brunt-Vaisala frequency
CD.	REAL*4	geostrophic drag coefficient
CHGTS . . .	REAL*4	height above local terrain
CLATM . . .	REAL*4	negative displaced latitude for wind calculations
CLATP . . .	REAL*4	positive displaced latitude for wind calculations

SIGD. . . .	REAL*4	standard deviation for perturbations (% of mean)
SINWLAT . .	REAL*4	sin of absolute value of latitude
SUNLAT. . .	REAL*4	latitude of sub-solar position
SUNLON . .	REAL*4	West longitude of sub-solar position
TAVG. . . .	REAL*4	daily average surface temperature
TAVM. . . .	REAL*4	dummy average surface temperature in ATMOS2 call
TAVP. . . .	REAL*4	dummy average surface temperature in ATMOS2 call
TCOS. . . .	REAL*4	COSPAR model atmospheric temperature
TFAC. . . .	REAL*4	dummy density perturbation factor for call to ATMOS2
TFACTOR . .	REAL*4	density perturbation factor at base of thermosphere
THGT. . . .	REAL*4	local terrain height
TLATM . . .	REAL*4	dummy temperature value for call to ATMOS2
TLATP . . .	REAL*4	dummy temperature value for call to ATMOS2
TLOCAL. . .	REAL*4	local time in "Martian hours" (1/24th Sols)
TLONM . . .	REAL*4	dummy temperature value for call to ATMOS2
TLONP . . .	REAL*4	dummy temperature value for call to ATMOS2
TMAM. . . .	REAL*4	dummy maximum surface temperature in ATMOS2 call
TMAP. . . .	REAL*4	dummy maximum surface temperature in ATMOS2 call
TMAX. . . .	REAL*4	daily maximum surface temperature
TMIM. . . .	REAL*4	dummy minimum surface temperature in ATMOS2 call
TMIN. . . .	REAL*4	daily minimum surface temperature
TMIP. . . .	REAL*4	dummy minimum surface temperature in ATMOS2 call
VAR	REAL*4	output height variable for the OUTPUT file
VARX. . . .	REAL*4	x (1st) variable for plotable output files
VARY. . . .	REAL*4	y (2nd) variable for plotable output files
VISC. . . .	REAL*4	coefficient of molecular viscosity
VISCFAC . .	REAL*4	viscous modification factor for winds
VLL	REAL*4	vertical scale for wind viscosity factor
VLS	REAL*4	vertical scale density for perturbations
WAVE. . . .	REAL*4	relative density perturbation from Zurek wave model
WIND. . . .	REAL*4	wind for use in perturbation magnitude
WLAT. . . .	REAL*4	latitude used for wind calculations
Z	REAL*4	height above ground (m), for boundary layer winds
Z0.	REAL*4	surface roughness (m)
Z1.	REAL*4	random variable, used in perturbation calculations
Z2.	REAL*4	random variable, used in perturbation calculations
ZF.	REAL*4	height of base of thermosphere
ZFACTOR . .	REAL*4	height factor in vertical interpolation
ZT.	REAL*4	dummy thermosphere base height, used in ATMOS2 call

Methodology:

Datastep is the principal driver subroutine for stepping through the positions and evaluating the atmospheric variables. It starts by reading the next position from the trajectory input file or by automatically generating the next position from the increments in latitude, longitude, altitude and time.

Terrain height is obtained by calling the Terrain subroutine (DSTP 34), orbit parameters are evaluated by calling the ORBIT subroutine (DSTP 36), dust storm perturbation factors (dustM and dustA) are evaluated by calling subroutine Dustfact (DSTP 38) and the other mean atmospheric parameters are evaluated by calling ATMOS2 (DSTP 40). High (+1 sigma) and low (-1 sigma) density values are computed from the factors FACTHI and FACTLO (DSTP 43 - DSTP 57). A normally-distributed random variate is found, using RANDOM and PPND (DSTP 58 - DSTP 61). The vertical scale for the wind viscosity factor is set to the pressure scale height (DSTP 65). The vertical and horizontal scales for the density perturbations are found from

```
VLS = 8.0                                             DSTP 69

HLS = 30. + .01875*CHGT**2                            DSTP 72
```

CLONM . . .	REAL*4	negative displaced longitude for wind calculations
CLONP . . .	REAL*4	positive displaced longitude for wind calculations
CORIOL. . .	REAL*4	Coriolis parameter for wind calculations
CORREL. . .	REAL*4	one-step correlation value for density perturbations
COSFAC. . .	REAL*4	cosine factor used in wind calculations
COSWLAT . .	REAL*4	cosine of absolute value of latitude
DAMPING . .	REAL*4	viscous damping factor in mountain wave perturbations
DATE. . . .	REAL*8	current Julian date
DCOS. . . .	REAL*4	COSPAR model atmospheric density
DELEW . . .	REAL*4	E-W displacement between previous and current position (DSTP 77); also E-W distance (= 5° longitude), for wind calculation (DSTP129)
DELNS . . .	REAL*4	N-S displacement between previous and current position (DSTP 76); also N-S distance (= 5° latitude), for wind calculation (DSTP120)
DELZ. . . .	REAL*4	height displacement, previous to current position
DEN0. . . .	REAL*4	dummy surface density argument for ATMOS2 subroutine
DENOM . . .	REAL*4	denominator term in wind calculations
DENSURF . .	REAL*4	atmospheric density at surface
DENSWA. . .	REAL*4	density wave perturbation amplitude, %
DEVAV . . .	REAL*4	deviation of average density from COSPAR value, %
DEVHI . . .	REAL*4	deviation of high (+1 sigma) density from COSPAR, %
DEVLO . . .	REAL*4	deviation of low (-1 sigma) density from COSPAR, %
DHGT. . . .	REAL*4	terrain height factor for mountain wave magnitudes
DLATM . . .	REAL*4	dummy density argument used in calling ATMOS2
DLATP . . .	REAL*4	dummy density argument used in calling ATMOS2
DLONM . . .	REAL*4	dummy density argument used in calling ATMOS2
DLONP . . .	REAL*4	dummy density argument used in calling ATMOS2
DMINUS. . .	REAL*4	difference between average and low density
DPLUS . . .	REAL*4	difference between high and average density
DUSTA . . .	REAL*4	relative dust storm magnitude for diurnal variations
DUSTM . . .	REAL*4	relative dust storm magnitude for daily mean values
FACTHI. . .	REAL*4	ratio of high (+1 sigma) to average density
FACTLO. . .	REAL*4	ratio of average to low (-1 sigma) density
FACTOR. . .	REAL*4	boundary layer wind factor
FHI	REAL*4	dummy high density factor for call to ATMOS2
FLO	REAL*4	dummy low density factor for call to ATMOS2
FUG	REAL*4	intermediate E-W component in wind calculations
FUGP. . . .	REAL*4	intermediate E-W component in wind calculations
FVG	REAL*4	intermediate N-S component in wind calculations
FVGP. . . .	REAL*4	intermediate N-S component in wind calculations
GZ.	REAL*4	acceleration of gravity at current height
HLATM . . .	REAL*4	dummy scale height argument for call to ATMOS2
HLATP . . .	REAL*4	dummy scale height argument for call to ATMOS2
HLONM . . .	REAL*4	dummy scale height argument for call to ATMOS2
HLONP . . .	REAL*4	dummy scale height argument for call to ATMOS2
HLS	REAL*4	horizontal scale for density perturbations
HSCALE. . .	REAL*4	pressure scale height
IFAULT. . .	INTEGER*4	error flag for the PPND function
L	INTEGER*4	error flag for the RANDOM function
MARSAU. . .	REAL*4	Sun-Mars orbital radius at current time
OHGT. . . .	REAL*4	height above reference ellipsoid (for output purposes)
OHGTS . . .	REAL*4	height above local terrain (for output purposes)
PCOS. . . .	REAL*4	COSPAR model atmospheric pressure
PERT. . . .	REAL*4	mountain wave perturbation magnitude
PERTMAX . .	REAL*4	maximum perturbation magnitude allowed by stability
PERTMIN . .	REAL*4	minimum allowed perturbation magnitude
PI180 . . .	REAL*4	factor to convert degrees to radians
PLATM . . .	REAL*4	pressure south of position, for pressure gradients
PLATP . . .	REAL*4	pressure north of position, for pressure gradients
PLONM . . .	REAL*4	pressure east of position, for pressure gradients
PLONP . . .	REAL*4	pressure west of position, for pressure gradients
RLATM . . .	REAL*4	dummy radius from Mars center, used in ATMOS2 call
RLATP . . .	REAL*4	dummy radius from Mars center, used in ATMOS2 call
RLONM . . .	REAL*4	dummy radius from Mars center, used in ATMOS2 call
RLONP . . .	REAL*4	dummy radius from Mars center, used in ATMOS2 call
RSC	REAL*4	radius from center of Mars to current position

```
      IF(HLS.GT.600.)HLS = 600.                                  DSTP 73
```

The components of the relative displacement between previous and current positions are computed

```
      DELNS = DTR*RSC*(DELLAT)/HLS                               DSTP 76
      DELEW = -DTR*RSC*COS(DTR*CLAT)*DELLON/HLS                  DSTP 77
      DELZ = DELHGT/VLS                                          DSTP 78
```

and used to evaluate the correlation between the previous and current position

```
      CORREL = Exp(-(Abs(DELNS) + Abs(DELEW) + Abs(DELZ)))       DSTP 81
```

Next, the wind components are computed from the horizontal pressure gradients (a Mars version of the geostrophic wind relations, referred to as the areostrophic winds; see discussion on computation of winds in the section "WIND PROFILES IN MARS-GRAM", of the Release 1 Technical Report, 1989; see Appendix B). The pressure gradients are computed by finite differences of the pressure across +/- 2.5° latitude and longitude displacements from the current position, by calling ATMOS2 (longitude + 2.5° at DSTP 86; longitude - 2.5° at DSTP 91; latitude + 2.5° at DSTP104 and latitude -2.5° at DSTP 113). The area-average terrain height, for use in the mountain-wave perturbation model, is evaluated with the Terrain function (DSTP 95 and DSTP117). For computing the pressure gradients, the distances corresponding to the 5 degree latitude or longitude displacements are computed at DSTP120 and DSTP129. The areostrophic wind components, computed from the pressure gradients are evaluated by

```
      FUG = -(PLATP-PLATM)/(DELNS*DENS)                          DSTP137
      FVG = (PLONP-PLONM)/(DELEW*DENS)                           DSTP138
```

[Release 1 Technical Report, equation (1); see Appendix B].

For modifications by molecular viscosity (relevant in the thermospheric region), viscosity factors are computed

```
      VISC = BETA*TEMP**1.5/(TEMP + SVAL)                        DSTP133
      VISCFAC = VISC/(1.0E6*DENS*VLL**2)                         DSTP134
```

For absolute latitudes less than 7.5°, a special low-latitude wind model [Release 1 Technical Report, equations (2) and (3); see Appendix B] is used

```
        FUGP = -4000.*RSC*SINWLAT*(PLATP + PLATM - 2.*PRES)/     DSTP141
     &   (DENS*COSWLAT*DELNS*DELNS)                              DSTP142
        FVGP = 0.0                                               DSTP143
```

For absolute latitudes between 7.5° and 15°, winds are interpolated between the low-latitude and regular values

```
          FUG = FUGP + (ABSWLAT-7.5)*(FUG-FUGP)/7.5              DSTP145
          FVG = FVGP + (ABSWLAT-7.5)*(FVG-FVGP)/7.5              DSTP146
```

For absolute latitudes greater than 75°, a special high-latitude relation [Release 1 Technical Report, equation (4); see Appendix B] is used

```
      Cosfac = COSWLAT/Cos(pi180*75.)                        DSTP154
      FUG = FUG*COSWLAT*Cosfac                               DSTP155
      FVG = FVG*COSWLAT*Cosfac                               DSTP156
```

Viscous-corrected wind components are computed by [Release 1 Technical Report, equations (6a) and (6b); see Appendix B]

```
     DENOM = CORIOL**2 + VISCFAC**2                          DSTP158

     EWWIND = (CORIOL*FUG - VISCFAC*FVG)/DENOM               DSTP160
     NSWIND =  (CORIOL*FVG + VISCFAC*FUG)/DENOM              DSTP161
```

For altitudes within 1 km of the surface, boundary-layer modifications are applied [Release 1 Technical Report, equations (8) and (9); see Appendix B]

```
      factor = (Sqrt(CD)/0.4)*Alog(z/z0)                     DSTP172
      if (factor .gt. 1.0) factor = 1.0                      DSTP173
      EWWIND = factor*EWWIND                                 DSTP174
      NSWIND = factor*NSWIND                                 DSTP175
```

where z is the height above the surface (m), the drag coefficient is taken to be CD = 0.0015 and the surface roughness is assumed to be z0 = 0.03 m.

For the mountain wave perturbation model [Release 1 Technical Report, equations (10) and (15); see Appendix B], the steps are:

(1) Compute the terrain height adjustment factor dhgt (local height above area-averaged terrain height, m) by

```
      dhgt = 1000.*(thgt - areahgt)                          DSTP184

      dhgt = 2.0*dhgt                                        DSTP187
      If (dhgt .lt. 10.)dhgt = 10.                           DSTP188
      If (dhgt .gt. 2000.)dhgt = 2000.                       DSTP189
```

(2) Compute the wind speed

```
      Wind = Sqrt(NSWIND**2 + EWWIND**2)                     DSTP191
      If (Wind .lt. 1.0)Wind = 1.0                           DSTP192
```

(3) Compute the viscous damping factor

```
      Damping = 1.0E6*HLS*VISC*(Bruntf**3)*Hscale*           DSTP194
     &  (1. - (DENS/densurf))/(12.56637*DENS*Wind**4)        DSTP195
      If (Damping .gt. 50.) Damping = 50.                    DSTP196
```

(4) Compute the perturbation magnitude and insure that it does not exceed the maximum magnitude allowed by stability and that it is at least as large as a prescribed minimum value (dependent on height)

```
      pert = FACTHI*dhgt*Exp(-Damping)                              DSTP198

      pertmax = (500.*VLS*Bruntf**2/gz)*(1. + 0.5*VLS/Hscale)       DSTP199c
      If (pertmax .gt. 0.5*VLS/Hscale) pertmax = 0.5*VLS/Hscale     DSTP199d
      If (pert .gt. pertmax)pert = pertmax                          DSTP200
      pertmin = 0.01 + 0.001*CHGT                                   DSTP201
      If (pertmin .gt. 0.05)pertmin = 0.05                          DSTP202
      If (pert .lt. pertmin)pert = pertmin                          DSTP203
```

(5) Between 75 km and the base of the thermosphere, interpolate between the mountain wave model and the magnitude derived from the Stewart thermosphere model at the base of the thermosphere

```
        zfactor = (CHGT - 75.0)/(ZF - 75.0)                         DSTP207
        pert = pert + zfactor*(tfactor - pert)                      DSTP208
```

(6) Values of DENSHI and DENSLO are computed, based on the mountain wave perturbation magnitudes

```
  585 DENSHI = DENS*(1. + pert)                                     DSTP212
      DENSLO = DENS/(1. + pert)                                     DSTP213
      DPLUS = DENSHI - DENS                                         DSTP214
      DMINUS = DENS - DENSLO                                        DSTP215
```

The local time, in Martian hours = 1/24th Sols is computed from the longitude of the current position

```
    TLOCAL = 12. + (SUNLON - CLON)/15.                              DSTP218
    IF (TLOCAL .LT. 0.)TLOCAL = TLOCAL + 24.                        DSTP219
    IF (TLOCAL .GT. 24.)TLOCAL = TLOCAL - 24.                       DSTP220
```

Height above the reference ellipsoid or above the local surface terrain height are evaluated and saved for output purposes

```
    OHGT = CHGT                                                     DSTP223
    OHGTS = CHGTS                                                   DSTP224
    IF(OHGT .LE. -5.)THEN                                           DSTP225
      OHGT = thgt                                                   DSTP226
      OHGTS = 0.                                                    DSTP227
    ENDIF                                                           DSTP228
```

The Zurek wave perturbation model (Section 4.3 of "The Mars Atmosphere: Observations and Model Profiles for Mars Missions", Report No. JSC-24455, David E. Pitts et al., eds., is used to add tidal wave perturbations

```
    If (modpert .ne. 1)Call Wavepert(OHGT,CLAT,TLOCAL,DustM,DustA,  DSTP231
   & wave,amp1,amp2)                                                DSTP232
```

The Zurek tidal wave model and the mountain-wave model perturbation are added, with the mountain wave values treated as having the previously-computed correlation between the prior and current positions

```
    RHO = CORREL*RHO + SQRT(1.0 - CORREL**2)*Z1                     DSTP235
    IF(RHO.LT.0.0)DENSP = DENS*(1. + wave) + RHO*DMINUS             DSTP236
```

```
IF(RHO.GE.0.0)DENSP = DENS*(1. + wave) + RHO*DPLUS              DSTP237
If (DENSP .lt. 0.0)DENSP = 0.05*DENS                            DSTP238
```

The standard deviation and DENSHI and DENSLO values are adjusted to reflect the Zurek wave perturbation values

```
SIGD = 50.*(DENSHI-DENSLO)/DENS                                 DSTP241

DENSHI = DENSHI + (amp1 + amp2)*DENS                            DSTP243
DENSLO = DENSLO - (amp1 + amp2)*DENS                            DSTP244

DENSP = 100.*(DENSP - DENS)/DENS                                DSTP246
```

Deviations from the COSPAR reference atmosphere are computed (DSTP254b to DSTP254k). Descriptively formatted data are written to the LIST file (DSTP248 and DSTP255). Output to the various plotable-output (graphics-input) files (units 21-34) are written, according to the options selected (NVARX, NVARY, and logscale; DSTP264b through DSTP349).

Subroutine: DZDUST **Code:** DZDS

Description: calculates the perturbation to ZF, the height of the base of the thermosphere, due to dust storms. Modified from the original Stewart subroutine, which assumed storms starting at Ls = 205 and Ls = 275 each year. Now uses the same starting Ls value as for lower atmosphere (if a dust storm case is selected).

Called By: ATMOS2, STEWART2

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
INTENS. . .	REAL*4	dust storm intensity level (0.0 - 3.0)
ALS0. . . .	REAL*4	starting Ls value for storm
LSUN. . . .	REAL*4	areocentric longitude of sun (Ls)

Output Variables (not passed through commons):

Name	Type	Description
DZ.	REAL*4	increase in height of base of thermosphere, due to dust storm, Stewart model method

Local Variables (not passed through commons):

Name	Type	Description
QA.	REAL*4	exponential decay factor for dust storm effect
DZA	REAL*4	intermediate value for DZ calculation
DLS	REAL*4	Ls duration for dust storm effect (25 degrees of Ls)
DLSA. . . .	REAL*4	difference in Ls value from start of storm (Ls - Ls0)

Methodology:

This subroutine has been modified from the original subroutine of Stewart (which still appears in the code, but commented out). Instead of assuming two storms per Mars year (one starting at Ls = 205, another starting at Ls = 275), DZDUST now uses the same Ls value for the beginning of the storm as is used for the lower altitude regions (based on user input value for Ls0).

The Ls difference, Ls-Ls0, is put in the range 0-25, by the relation

```
DLSA = AMOD(((LSUN - als0) + 720.0), 360.0)                DZDS 27
```

The exponential time-decay factor is computed by

```
QA = EXP(-DLSA / DLS)                                      DZDS 28
```

and the final displacement for the base of the thermosphere (km) is computed by

```
DZA = 5.0*INTENS                                                    DZDS 29
DZ = DZA*QA*(1.0 - QA**4)*1.869                                     DZDS 30
```

Subroutine: EScalc **Code:** ESCL

Description: Computes short-term and long-term standard deviations from nominal densities in Stewart thermosphere model.

Called By: ATMOS2, STEWART2

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
SIGMA. . .	REAL*4	long-term thermospheric variability factor
STDL . . .	REAL*4	short-term thermospheric variability factor

Output Variables (not passed through commons):

Name	Type	Description
ES	REAL*4	array for thermospheric variability factors

Local Variables (not passed through commons):

Name	Type	Description
I.	INTEGER*4	index for the ES array
EPS. . . .	REAL*4	intermediate array for ES calculations
SIG. . . .	REAL*4	array of coefficients for ES calculations

Methodology:

The short-term thermospheric variability factor, stdl, is read in the main (MarsGRAM) or SETUP routines. The long-term variability factor, SIGMA, is set by the calling value in the STEWART2 argument list (e.g. 0.0 at ATM2143, 1.0 at ATM2146 and ATM2164, or -1.0 at ATM2166).

Even-number indexed ES values (0, 2, 4, 6, 8, 10) are for long-term variability; odd-number indexed ES values (1, 3, 5, 7, 9, 11) are for short-term variability. ES(0) and ES(1) are for the dependence on solar activity (FBAR at STW2 25). ES(2) and ES(3) are for the exospheric temperature (TINF at STW2 46). ES(4) through ES(7) are for the atomic oxygen factors (FO at THRM 38 and AO at THRM 36). ES(8) and ES(9) are for the base-height of the thermosphere (ZF at ATM2137 and STW2 38) and the temperature at the base of the thermosphere (TF at STW2 47). ES(10) and ES(11) are for the dust storm effect on the thermosphere (DUST at ATM2135 and STW2 37).

The factors ES(1), ES(6) and ES(11) are set to zero in the Stewart thermosphere model, and have therefore been left out of the equations that they would otherwise appear in (i.e. they do not appear explicitly in lines STW2 25 [ES(1)], THRM 36 [ES(6)] or in ATM2135 or STW2 37 [ES(11)].

Even-number indexed values of EPS are set to stdl; odd-number indexed values of EPS are set to SIGMA. The SIG array elements are set to assigned values and ES is computed as the product of EPS and SIG.

Subroutine: gamma **Code:** GAMA

Description: Computes lapse rates (-dT/dz) over the significant level areopotential height intervals, for a given latitude and Ls value.

Called By: ATMOS2

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
ALAT. . . .	REAL*4	current latitude
ALS	REAL*4	areocentric longitude of the sun (Ls)

Output Variables (not passed through commons):

Name	Type	Description
GAM	REAL*4	lapse rate array (negative of temperature gradient)

Local Variables (not passed through commons):

Name	Type	Description
A	REAL*4	coefficient for part of gamma depending only on latitude
A1.	REAL*4	gam(1) (0-5 km) coefficients for A term
A2.	REAL*4	gam(2) (5-15 km) coefficients for A term
A3.	REAL*4	gam(3) (15-30 km) coefficients for A term
B	REAL*4	coefficient depending on latitude and sine of Ls
B1.	REAL*4	gam(1) (0-5 km) coefficients for B term
B2.	REAL*4	gam(2) (5-15 km) coefficients for B term
B3.	REAL*4	gam(3) (15-30 km) coefficients for B term
C	REAL*4	coefficient depending on latitude and cosine of Ls
C1.	REAL*4	gam(1) (0-5 km) coefficients for C term
C2.	REAL*4	gam(2) (5-15 km) coefficients for C term
C3.	REAL*4	gam(3) (15-30 km) coefficients for C term
CALS. . . .	REAL*4	cosine of Ls angle
F180. . . .	REAL*4	array of Fourier terms for $\pi/90$ times latitude
F240. . . .	REAL*4	array of Fourier terms for $\pi/120$ times latitude
I	INTEGER*4	index value
PI120 . . .	REAL*4	$\pi/120$
PI180 . . .	REAL*4	$\pi/180$
PI90. . . .	REAL*4	$\pi/90$
SALS. . . .	REAL*4	sine of Ls angle

Methodology:

Uses Fourier coefficients for (sine and cosine) terms in latitude and Ls angle. The temperature lapse rate (-dT/dz) array, gam (in units of K/km), has 5 elements, one for each height interval between the significant levels: gam(1) applies to heights 0-5 km, gam(2) is for 5-15 km, gam(3) is for 15-30 km, gam(4) is for 30-50 km, and gam(5) applies to 50-75 km. The upper two intervals are assumed to have fixed lapse rates: gam(4) = 0.9 K/km and gam(5) = 0.4 K/km. For the three lower levels, the gam values are computed by an equation of the form

$$\text{gam} = A + B \sin(Ls) + C \cos(Ls)$$

(GAMA 58, GAMA 68 and GAMA 78).

The A coefficient term is computed as a Fourier series, in terms of sines and cosines of multiples of $\pi/90$ times the latitude (GAMA 54, GAMA 64, and GAMA 74). The B and C coefficient terms are computed as Fourier series, in terms of sines and cosines of multiples of $\pi/120$ times the latitude (GAMA 55, GAMA 57, GAMA 65, GAMA 67, GAMA 75 and GAMA 77). Coefficients (A1, A2, A3, B1, B2 B3, C1, C2, and C3 arrays) are stored in data statements, indexed 0 through 8, for the constant coefficient and the 8 Fourier terms in the respective latitude series.

Subroutine: orbit **Code:** ORBT

Description: Computes orbital radius of Mars from Sun, areocentric longitude of the sun (Ls) and the latitude and longitude of the sub-solar point on the Mars surface for a given time (expressed as a Julian date).

Called By: MarsGRAM (Main), Datastep, SETUP

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
XDATE . . .	REAL*8	Julian date for current time

Output Variables (not passed through commons):

Name	Type	Description
LATSUN. . .	REAL*4	latitude of sub-solar point on Mars on Mars surface
LONSUN. . .	REAL*4	West longitude of sub-solar point on Mars surface
LSUBS . . .	REAL*4	areocentric longitude of the sun (like right ascension)
RADIUS. . .	REAL*4	magnitude of vector from Sun to orbital position of Mars

Local Variables (not passed through commons):

Name	Type	Description
A0 thru A6	REAL*8	Fourier coefficients for RADIUS calculation
B0 thru B6	REAL*8	Fourier coefficients for LATSUN calculation
C0 thru C6	REAL*8	Fourier coefficients for LSUBS calculation
D0 thru D6	REAL*8	Fourier coefficients for LONSUN calculation
DATE. . . .	REAL*8	modified Julian date (xdate - 2442779. + 0.5)
DS.	REAL*8	intermediate value of LATSUN used for calculation
LON	REAL*8	intermediate value of LONSUN used for calculation
LON0. . . .	REAL*8	initial value of LONSUN
LS.	REAL*8	intermediate value of LSUBS used for calculation
LS0	REAL*8	initial value of LSUBS
PER1. . . .	REAL*8	687 day period (used for LATSUN, LSUBS and RADIUS)
PER2. . . .	REAL*8	696 day period (used for LONSUN)
PERLON. . .	REAL*8	coefficient for linear increase of LONSUN with date
PERLS . . .	REAL*8	coefficient for linear increase of LSUBS with date
PI180 . . .	REAL*8	π / 180
RAD	REAL*8	intermediate value of RADIUS used for calculation
TIME1 . . .	REAL*8	current time variable with period = PER1
TIME2 . . .	REAL*8	current time variable with period = PER2
TWOPI . . .	REAL*8	2 π
XLON. . . .	REAL*8	part of LONSUN increasing linearly with date
XLS	REAL*8	part of LSUBS increasing linearly with date

Methodology:

Uses pre-computed coefficients in Fourier time series to compute LATSUN, LONSUN, LSUBS and RADIUS from the modified Julian date

```
      DATE = XDATE - 2.442779d6 + 0.5d0                                     ORBT 18
```

The parts of LSUBS and LONSUN that increase linearly with the date are computed from

```
      xls = ls0 + 3.60d2*date/perls                                         ORBT 19
      xlon = lon0 + 3.60d2*(date - 2.922d3)/perlon                          ORBT 20
```

The time variables for period 1 (687 day period) and period 2 (696 day period) are evaluated by

```
      TIME1=DATE/PER1                                                       ORBT 21
      TIME2=DATE/PER2                                                       ORBT 22
```

LSUBS (intermediate value LS), in degrees, is computed from

```
      LS=C0+C1*DSIN(TIME1)+C2*DCOS(TIME1)+C3*DSIN(2.0d0*TIME1)              ORBT 23
     & +C4*DCOS(2.0d0*TIME1)+C5*DSIN(3.0d0*TIME1)+C6*DCOS(3.0d0*TIME1)      ORBT 24
     & +XLS                                                                 ORBT 25
```

LATSUN (intermediate value DS), in degrees, is computed from

```
      DS=B0+B1*DSIN(TIME1)+B2*DCOS(TIME1)+B3*DSIN(2.0d0*TIME1)              ORBT 26
     & +B4*DCOS(2.0d0*TIME1)+B5*DSIN(3.0d0*TIME1)+B6*DCOS(3.0d0*TIME1)      ORBT 27
```

LONSUN (intermediate value LON), in degrees, is computed from

```
      LON=D0+D1*DSIN(TIME2)+D2*DCOS(TIME2)+D3*DSIN(2.0d0*TIME2)             ORBT 28
     & +d4*DCOS(2.0d0*time2)+d5*DSIN(3.0d0*time2)+d6*DCOS(3.0d0*time2)      ORBT 29
     & +xlon                                                                ORBT 30
```

RADIUS (intermediate value RAD), in astronomical units (AU), is computed from

```
      rad = a0 + a1*DSIN(time1) + a2*DCOS(time1)                            ORBT 35
     & + a3*DSIN(2.0d0*time1) + a4*DCOS(2.0d0*time1)                        ORBT 36
     & + a5*DSIN(3.0d0*time1) + a6*DCOS(3.0d0*time1)                        ORBT 37
```

The Fourier coefficients (A0-A6, B0-B6, C0-C6 and D0-D6) are loaded in via data statements. Both LSUBS and LONSUN are put into the range 0 to 360 degrees.

Function: phasint **Code:** PHSN

Description: Linear interpolation on phase angle, by vector components, for the phases of the wave perturbation field.

Called By: Wavepert

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
DX.	REAL*4	linear interpolation value (0-1)
OMEGA . . .	REAL*4	frequency for the phase angle interpolation
PH1	REAL*4	phase angle at first position (location where DX=0)
PH2	REAL*4	phase angle at second position (location where DX=1)

Output Variables (not passed through commons):

Name	Type	Description
PHASINT . .	REAL*4	interpolated phase angle at time corresponding input DX

Local Variables (not passed through commons):

Name	Type	Description
DPH	REAL*4	phi2 - phi1
X	REAL*4	interpolated "x" vector component of phase
X1.	REAL*4	"x" component of phase at time 1 [sin(omega*phi1)]
X2.	REAL*4	"x" component of phase at time 2 [sin(omega*phi2)]
Y	REAL*4	interpolated "y" vector component of phase
Y1.	REAL*4	"y" component of phase at time 1 [cos(omega*phi1)]
Y2.	REAL*4	"y" component of phase at time 2 [cos(omega*phi2)]

Methodology:

This routine is used to interpolate wave phase angle as a function of latitude or height, in the Zurek wave model (subroutine Wavepert). It treats phase angle as a circular variable, having "x" and "y" Cartesian vector components. The input parameter omega is the circular frequency for the latitude or height variation. Input values of phase angle (phi1 and phi2) corresponding to x and y component values

```
x1 = sin(omega*ph1)                                  PHSN  4
x2 = sin(omega*ph2)                                  PHSN  5
y1 = cos(omega*ph1)                                  PHSN  6
y2 = cos(omega*ph2)                                  PHSN  7
```

The interpolated values of the vector components are

```
x = x1 + (x2 - x1)*dx                                PHSN  8
y = y1 + (y2 - y1)*dx                                PHSN  9
```

The interpolated value of the phase angle is reconstructed from the vector components by

```
phasint = atan2(x,y)/omega                                PHSN 10
```

The routine is also used to interpolate linearly on 24 hour local time variables, by the relations

```
dph = ph2 - ph1                                           PHSN 12
If (abs(dph) .gt. 12.)dph = dph - sign(24.,dph)           PHSN 13
phasint = ph1 + dph*dx                                    PHSN 14
```

Function: polecap **Code:** POLC

Description: Computes radius (latitude degrees) of the edge of the polar cap from the pole, for a given areocentric longitude of sun (Ls), for either north polar cap or south polar cap.

Called By: Alb, Tsurface

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
ALAT. . . .	REAL*4	latitude (+ for north pole, - for south pole)
ALS	REAL*4	areocentric longitude of the sun (Ls)

Output Variables (not passed through commons):

Name	Type	Description
POLECAP . .	REAL*4	polar cap radius (degrees of latitude)

Local Variables (not passed through commons):

Name	Type	Description
ALS0. . . .	REAL*4	polar cap phase in Ls angle
PI180 . . .	REAL*4	π / 180

Methodology:

Uses a sine wave in Ls to approximate the latitude progression and regression of the polar caps. The radial distance (in degrees of latitude) from the pole to the cap edge is computed by

```
polecap = 19. - 16.*sin(pi180*(als-als0))                    POLC 10
```

where als0 = 230 for the north polar cap and als0 = 50 for the south polar cap.

Function: PPND **Code:** PPND

Description: Produces a normally-distributed deviate, for use in the random perturbation model.

Called By: Datastep

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
p	REAL*4	input value, from uniformly distributed random number

Output Variables (not passed through commons):

Name	Type	Description
IFAULT. . .	INTEGER*4	error flag
PPND. . . .	REAL*4	normally (Gaussian) distributed random variable

Local Variables (not passed through commons):

Name	Type	Description
A0 thru A3	REAL*4	coefficients used in computing PPND
B1 thru B4	REAL*4	coefficients used in computing PPND
C0 thru C3	REAL*4	coefficients used in computing PPND
D1 thru D2	REAL*4	coefficients used in computing PPND
HALF. . . .	REAL*4	1./2.
ONE	REAL*4	1.
Q	REAL*4	p - 1./2.
R	REAL*4	the square of Q
SPLIT . . .	REAL*4	0.42
ZERO. . . .	REAL*4	0.

Methodology:

Uses the method of algorithm AS 111, Applied Statistics (1977), Volume 26, page 118. The empirical coefficients (A0 through C3) are used to fit the normally-distributed value PPND corresponding to the lower tail area p of the normal error integral. The error flag ifault is set to 0 if 0 < p < 1. Otherwise, ifault is set to 1 and PPND is set to 0.

Subroutine: Pressure **Code:** PRES

Description: Computes an array of atmospheric pressures at the significant height levels, from the surface pressure, the terrain height, surface gravity, local radius of reference ellipsoid, and temperatures at the significant levels. Also computes the areographic altitudes, z(i), corresponding to the areopotential altitudes, gph(i), for the significant levels.

Called By: ATMOS2

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
GAM	REAL*4	array of lapse rates across the significant levels
GZERO . . .	REAL*4	surface gravity
PSURF . . .	REAL*4	surface pressure
RBAR. . . .	REAL*4	gas constant
RREF. . . .	REAL*4	local radius of reference ellipsoid
T	REAL*4	array of temperatures at the significant levels
THGT. . . .	REAL*4	terrain height above reference ellipsoid

Output Variables (not passed through commons):

Name	Type	Description
P	REAL*4	array of pressures at the significant levels
Z	REAL*4	array of areographic heights at the significant levels

Local Variables (not passed through commons):

Name	Type	Description
I	INTEGER*4	array index
GOR	REAL*4	gravity divide by gas constant
DH.	REAL*4	height increment array between significant levels
GPH	REAL*4	areopotential height array of the significant levels

Methodology:

Evaluates the areographic altitudes, z(i), that correspond to the areopotential heights, gph(i), of the significant levels (0, 5, 15, 30, 50 and 75 areopotential km).

```
z(i) = (gph(i)+thgt)/(1. - (gph(i)+thgt)/rref)                PRES 22
```

Evaluates the pressure array p(i) at the significant level altitudes, using

```
p(i) = p(i-1)*exp(-goR*alog(T(i-1)/T(i))/gam(i))              PRES 25
```

if the lapse rate, gam(i), is not zero, or

```
p(i) = p(i-1)*exp(-goR*dh(i)/T(i-1))
```
PRES 27

if the lapse rate is zero.

Subroutine: PRSEAS **Code:** PRSE

Description: Computes the relative seasonal variation in atmospheric pressure on the reference ellipsoid, at a given latitude and areocentric longitude of the sun (Ls).

Called By: ATMOS2, Psurface, STEWART2

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
LAT	REAL*4	latitude of current position
LSUN. . . .	REAL*4	areocentric longitude of the sun (Ls)

Output Variables (not passed through commons):

Name	Type	Description
PR.	REAL*4	relative pressure variation with latitude and season at height of reference ellipsoid

Local Variables (not passed through commons):

Name	Type	Description
A1.	REAL*4	cos(Ls) coefficient
A11	REAL*4	constant coefficient of A1 term
A12	REAL*4	latitude coefficient of A1 term
A2.	REAL*4	cos(2 Ls) coefficient
A21	REAL*4	constant coefficient of A2 term
A22	REAL*4	latitude coefficient of A1 term
LAT2. . . .	REAL*4	square of latitude
PHI1. . . .	REAL*4	Ls phase angle of cos(Ls) term
PHI11 . . .	REAL*4	constant coefficient of PHI1 term
PHI12 . . .	REAL*4	latitude coefficient of PHI1 term
PHI2. . . .	REAL*4	Ls phase angle of cos(2 Ls) term
PHI21 . . .	REAL*4	constant coefficient of PHI2 term
PHI22 . . .	REAL*4	latitude coefficient of PHI2 term
PI180 . . .	REAL*4	π / 180

Methodology:

Computes amplitudes of cos(Ls) and cos(2 Ls) terms as a function of the square of the latitude by

```
a1 = a11 + a12*Lat2                                    PRSE 10
a2 = a21 + a22*Lat2                                    PRSE 11
```

Computes Ls phase angles of cos(Ls) and cos(2 Ls) terms by

```
phi1 = phi11 + phi12*Lat2                              PRSE 12
phi2 = phi21 + phi22*Lat2                              PRSE 13
```

Compute the relative pressure, as a function of latitude and Ls = Lsun by

```
      PR = 1. + a1*cos(pi180*(Lsun - phi1)) +                          PRSE 14
     & a2*cos(2.*pi180*(Lsun - phi2))                                  PRSE 15
```

Subroutine: Psurface **Code:** PSRF

Description: Computes the daily average surface pressure and local surface pressure at a given latitude and longitude for a given areocentric longitude of the sun (Ls), and daily average surface temperature.

Called By: ATMOS2

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
ALS	REAL*4	areocentric longitude of the sun (Ls)
DUSTA . . .	REAL*4	dust storm effect of diurnal amplitude
DUSTM . . .	REAL*4	dust storm effect on daily mean values
GAMMA . . .	REAL*4	local temperature lapse rate (K/km)
GZERO . . .	REAL*4	surface gravity
RBAR. . . .	REAL*4	gas constant
SITLA . . .	REAL*4	local site latitude
SITLO . . .	REAL*4	local site longitude
SUNLON. . .	REAL*4	longitude of sub-solar point
TAVG. . . .	REAL*4	daily average surface temperature
THGT. . . .	REAL*4	local terrain height

Output Variables (not passed through commons):

Name	Type	Description
PAVG. . . .	REAL*4	array of daily average pressures at significant levels
PSURF . . .	REAL*4	daily average local surface pressure

Local Variables (not passed through commons):

Name	Type	Description
A11	REAL*4	diurnal amplitude coefficient
A12	REAL*4	diurnal amplitude coefficient
A21	REAL*4	semi-diurnal amplitude coefficient
A22	REAL*4	semi-diurnal amplitude coefficient
ABSLAT. . .	REAL*4	absolute value of latitude
ALSM60. . .	REAL*4	Ls - 60 degrees, converted to radians
AMP1. . . .	REAL*4	amplitude of diurnal dust effect
AMP2. . . .	REAL*4	amplitude of semi-diurnal dust effect
C0 thru C2	REAL*4	coefficients in pressure/temperature ratio
COSLAT. . .	REAL*4	cosine of latitude
COSLS . . .	REAL*4	cosine of Ls - 60
D0 thru D2	REAL*4	coefficients in pressure/temperature ratio
EXPONENT. .	REAL*4	exponent in pressure variation with temperature
FACTOR. . .	REAL*4	coefficient in latitude and seasonal daily pressure
FREQ. . . .	REAL*4	diurnal frequency (15 degrees per Mars hour)
H11	REAL*4	diurnal phase coefficient
H12	REAL*4	diurnal phase coefficient
H21	REAL*4	semi-diurnal phase coefficient
H22	REAL*4	semi-diurnal phase coefficient
NMALS . . .	REAL*4	square of 90 - absolute latitude
NMLS. . . .	REAL*4	square of 90 - latitude
NPLS. . . .	REAL*4	square of 90 + latitude
PH1	REAL*4	time phase of diurnal surface pressure variation

PH2	REAL*4	time phase of semi-diurnal surface pressure variation
PI180 . . .	REAL*4	π / 180
POTAVG. . .	REAL*4	annual average pressure/temperature ratio
PR.	REAL*4	seasonally and latitudinally dependent relative pressure
PREF. . . .	REAL*4	daily average temperature on reference ellipsoid
RREF. . . .	REAL*4	local radius of reference ellipsoid
SINLS . . .	REAL*4	sine of Ls
TBAR. . . .	REAL*4	estimated annual average temperature versus latitude
TLOCAL. . .	REAL*4	local time in Mars hours (1/24th Sols)
TREF. . . .	REAL*4	temperature in conversion of pressure to terrain height
XLAT. . . .	REAL*4	current latitude in radians

Methodology:

Compute local radius of reference ellipsoid and surface gravity at current latitude, using the RELLIPS subroutine. Evaluates the annual average of the pressure/temperature ratio on the reference ellipsoid

```
      If (abslat .lt. 55.) then                                    PSRF 29
        potavg = 0.5*(2.*C0 + C1*(npls+nmls) + C2*(npls**2         PSRF 30
     &   +nmls**2))                                                PSRF 31
      Else                                                         PSRF 32
        potavg = 0.5*(D0 + D1*abslat + D2*abslat**2 +              PSRF 33
     &   C0 + C1*nmals + C2*nmals**2)                              PSRF 34
      Endif                                                        PSRF 35
```

Computes the annual average temperature as a function of latitude

```
      Tbar = (((0.525161E-6*sitla - 0.263317E-5)*sitla             PSRF 37
     & - 0.0116584)*sitla + 0.0334196)*sitla + 216.54              PSRF 38
```

Uses the PRSEAS subroutine to get the relative pressure for the given latitude and Ls value, then converts this to actual daily average pressure at the height of the reference ellipsoid

```
      Call PRSEAS(als,sitla,pr)                                    PSRF 40

      factor = 1.3854 - 3.59437E-5*sitla**2                        PSRF 43
      pref = factor*pr*potavg*Tbar                                 PSRF 44
```

Converts pressure on the reference ellipsoid to pressure on the local terrain surface

```
      If (abs(gamma) .lt. 0.001) then                              PSRF 46
        pavg = pref*exp(-1000.*gzero*thgt/(Rbar*Tavg))             PSRF 47
      Else                                                         PSRF 48
        Tref = Tavg + gamma*thgt                                   PSRF 49
        exponent = -1000.*gzero/(Rbar*gamma)                       PSRF 50
        pavg = pref*((Tref/Tavg)**exponent)                        PSRF 51
      Endif                                                        PSRF 52
```

Computes the effects of dust storm (if any) on the diurnal average pressure

```
      pavg = pavg + (61.082*sin(2.*pi180*sitla)                    PSRF 54
     & - 10.837*sin(4.*pi180*sitla))*dustM                         PSRF 55
```

Evaluates the local time in Mars hours

```
      tlocal = 12. + (sunlon - sitlo)/15.                               PSRF 57
      If (tlocal .lt. 0.) tlocal = tlocal + 24.                         PSRF 58
      If (tlocal .gt. 24.)tlocal = tlocal - 24.                         PSRF 59
```

Computes the amplitude (amp1, N/m^2) and phase (ph1, hours) of the diurnal variation in surface pressure

```
      sinLs = sin(pi180*als)                                            PSRF 61
      A11 = 12.60 - 0.118*sitla                                         PSRF 62
      A12 = -0.60 - 0.063*sitla                                         PSRF 63
      amp1 = A11 + A12*sinLs                                            PSRF 64
      H11 = 7.769 - 0.05769*sitla                                       PSRF 65
      H12 = 3.077 + 0.001923*sitla                                      PSRF 66
      ph1 = H11 - H12*sinLs                                             PSRF 67
```

Computes the amplitude (amp2, N/m**2) and phase (ph2, hours) of the semi-diurnal variation in surface pressure

```
      A21 = 7.982 - 0.022*sitla                                         PSRF 69
      A22 = -5.219 + 0.033*sitla                                        PSRF 70
      amp2 = A21 + A22*sinLs                                            PSRF 71
      H21 = 11.635 - 0.02885*sitla                                      PSRF 72
      H22 = -0.135 + 0.02885*sitla                                      PSRF 73
      alsm60 = pi180*(als-60.)                                          PSRF 74
      cosLs = cos(alsm60)                                               PSRF 75
      ph2 = H21 + H22*cosLs                                             PSRF 76
```

Adds the dust storm effects on the diurnal and semi-diurnal amplitudes

```
      amp1 = amp1 + (1.32 + 13.56*coslat)*dustA                         PSRF 80
      amp2 = amp2 + (33.33*exp(-.618727E-3*sitla**2))*dustA             PSRF 81
```

Evaluates the local surface pressure as the daily average value plus the diurnal and semi-diurnal variations of pressure about the daily average

```
      psurf = pavg + amp1*cos(freq*(tlocal-ph1))                        PSRF 85
     & + amp2*cos(2.*freq*(tlocal-ph2))                                 PSRF 86
```

Function: Random **Code:** RAND

Description: Returns a pseudo-random number, rectangularly distributed between 0 and 1.

Called By: MarsGRAM (Main), Datastep, SETUP

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

RANDCOM

Input Variables (not passed through commons):

Name	Type	Description
None		

Output Variables (not passed through commons):

Name	Type	Description
L	INTEGER*4	error flag
RANDOM. . .	REAL*4	uniformly distributed random variate (0 - 1)

Local Variables (not passed through commons):

Name	Type	Description
ONE	REAL*4	1.0
ZERO. . . .	REAL*4	0.0

Methodology:

Returns a pseudo random variate, uniformly distributed between 0 and 1, using the method of Algorithm AS 183, Applied Statistics (1982), Volume 31, page 188. Uses the integer variables IX, IY and IZ, passed through common RANDCOM. Successive values of IX, IY and IZ are computed by

```
IX = 171 * Mod(IX, 177) -  2 * (IX / 177)                RAND 19
IY = 172 * Mod(IY, 176) - 35 * (IY / 176)                RAND 20
IZ = 170 * Mod(IZ, 178) - 63 * (IZ / 178)                RAND 21

If (IX .lt. 0) IX = IX + 30269                           RAND 23
If (IY .lt. 0) IY = IY + 30307                           RAND 24
If (IZ .lt. 0) IZ = IZ + 30323                           RAND 25
```

The pseudo-random deviate value is computed from

```
 Random = Amod(float(IX) / 30269.0 + float(IY) / 30307.0 +    RAND 37
& float(IZ) / 30323.0, one)                                    RAND 38
```

Subroutine: RELLIPS **Code:** RLPS

Description: Calculates the reference radius of Cain et al's 6.1 mbar reference ellipsoid, for a given latitude. Also computes the acceleration of gravity for a given altitude above the reference ellipsoid at that latitude.

Called By: MarsGRAM (Main), ATMOS2, Datastep, Psurface, STEWART2, THERMOS, SETUP

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
LAT	REAL*4	current latitude (degrees)
Z	REAL*4	current height (km)

Output Variables (not passed through commons):

Name	Type	Description
GZ.	REAL*4	acceleration of gravity at height z (m/s^2)
RREF. . . .	REAL*4	local radius of reference ellipsoid (km)

Local Variables (not passed through commons):

Name	Type	Description
A	REAL*4	largest equatorial radius of reference ellipsoid
AB.	REAL*4	square root of product of A and B
B	REAL*4	smallest equatorial radius of reference ellipsoid
C	REAL*4	polar radius of reference ellipsoid
GM.	REAL*4	gravitational constant time mass of Mars
J2.	REAL*4	coefficient of 1st non-spherical component of gravity
P2.	REAL*4	latitude part of non-spherical component of gravity
RADEG . . .	REAL*4	180 / π
RZ.	REAL*4	radius to current height
TLAT. . . .	REAL*4	tangent of latitude
XX.	REAL*4	square of x component of local ellipsoid radius
YY.	REAL*4	square of y component of local ellipsoid radius

Methodology:

Uses geometry of an ellipsoid and the reference radii A = 3394.67 km, B = 3393.21 km, and C = 3376.78 to compute the local radius of the reference ellipsoid

```
XX = (AB * C)**2 / (C**2 + (AB * TLAT)**2)                    RLPS 15
YY = XX * TLAT**2                                             RLPS 16
Rref = SQRT(XX + YY)                                          RLPS 17
```

Compute the acceleration of gravity at altitude z, by the method of Seiff (Advances in Space Research, Volume 2, 1982, pages 3-17). This includes the spherically symmetric gravitational component and the first (J2) non-spherically symmetric component

```
      GM = 4.28282E7                                           RLPS 19
      J2 = 0.001965                                            RLPS 20
      P2 = 1.5*sin(LAT / RADEG)**2 - 0.5                       RLPS 21
      Rz = Rref + z                                            RLPS 22
      gz = (GM/Rz**2)*(1. - 3.*J2*((AB/Rz)**2)*P2)             RLPS 23
```

Subroutine: Stratos **Code:** STRA

Description: Interpolates between z75, the areographic height for the 75 km areopotential significant level, and ZF, the base of the Stewart thermosphere. Computes temperature, pressure, density, scale height, and temperature gradient at a given height z between these two levels.

Called By: ATMOS2

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
GZ.	REAL*4	acceleration of gravity at current altitude
M75	REAL*4	molecular weight at 75 km areopotential km
MF.	REAL*4	molecular weight at base of thermosphere
P75	REAL*4	pressure at 75 km areopotential km
PF.	REAL*4	pressure at base of thermosphere
RREF. . . .	REAL*4	local radius of reference ellipsoid
RSTAR . . .	REAL*4	gas constant
T50	REAL*4	temperature at 50 km areopotential km
T75	REAL*4	temperature at 75 km areopotential km
TF.	REAL*4	temperature at base of thermosphere
Z	REAL*4	current altitude (km)
Z50	REAL*4	areographic height at 50 km areopotential km
Z75	REAL*4	areographic height at 75 km areopotential km
ZF.	REAL*4	height at base of thermosphere

Output Variables (not passed through commons):

Name	Type	Description
DZ.	REAL*4	atmospheric density at current altitude
HZ.	REAL*4	scale height at current altitude
PZ.	REAL*4	atmospheric pressure at current altitude
TGRAD . . .	REAL*4	vertical temperature gradient (K/km)
TZ.	REAL*4	atmospheric temperature at current altitude

Local Variables (not passed through commons):

Name	Type	Description
DH.	REAL*4	height interpolation factor (H - H75)/(HF - H75)
EXPON . . .	REAL*4	exponent in pressure versus temperature dependence
GAM1. . . .	REAL*4	temperature gradient between H75 and HM
GAM2. . . .	REAL*4	temperature gradient between HM and HF
GOR	REAL*4	gravity divided by gas constant
H	REAL*4	areopotential altitude corresponding to Z
H50	REAL*4	areopotential altitude corresponding to Z50
H75	REAL*4	areopotential altitude corresponding to Z75
HF.	REAL*4	areopotential altitude corresponding to ZF
HFP	REAL*4	HF adjusted by molecular weight ratio
HM.	REAL*4	midpoint between H75 and HFP
HP.	REAL*4	H adjusted by molecular weight ratio
M	REAL*4	molecular weight at height Z
PM.	REAL*4	pressure at midpoint height HM
TM.	REAL*4	temperature at midpoint height HM

Methodology:

Uses input values of z75 and z50, the areographic altitudes above the reference ellipsoid corresponding to 75 km and 50 km areopotential altitude above local surface, to compute H75 and H50, the areopotential altitudes above the reference ellipsoid, corresponding to z75 and Z50

```
H75 = z75/(1. + z75/rref)                                   STRA  8
H50 = z50/(1. + Z50/rref)                                   STRA  9
```

Finds HF, the areopotential altitude corresponding to zF, areographic altitude of the base of the thermosphere

```
HF = zF/(1. + zF/rref)                                      STRA 12
```

Computes H, the areopotential altitude corresponding to the current areographic altitude z, relative to the reference ellipsoid

```
H = z/(1. + z/rref)                                         STRA 15
```

Interpolates on height to get M, the molecular weight at height z

```
dH = (H - H75)/(HF - H75)                                   STRA 17
M = M75 + (MF - M75)*dH                                     STRA 18
```

Adjusts the heights HF and H by the molecular weight ratio

```
HFP = MF*HF/M75                                             STRA 20
HP = M*H/M75                                                STRA 21
```

Finds HM, the midpoint (molecular weight adjusted) height between z75 and ZF

```
HM = 0.5*(HFP + H75)                                        STRA 23
```

Uses quadratic height interpolation (fitting the temperature at heights H50, H75 and HFP) to get the temperature TM at the midpoint height HM

```
 TM = -0.25*T50*(HFP-H75)**2/((H75-H50)*(HFP-H50))          STRA 25
&  + 0.5*T75*(HM-H50)/(H75-H50) + 0.5*TF*(HM-H50)/(HFP-H50) STRA 26
 If (TM.eq.T75)TM = T75 + 1.                                STRA 27
 If (TM.eq.TF)TM = TF - 1.                                  STRA 28
```

Computes the temperature gradients for the two layers H75 to HM and HM to HF

```
gam1 = (TM - T75)/(HM - H75)                                STRA 30
gam2 = (TF - TM)/(HFP - HM)                                 STRA 31
```

Find the exponent factors for pressure variation in the two layers

```
      goR = Alog(p75/pF)/(Alog(TM/T75)/gam1 + Alog(TF/TM)/gam2)         STRA 33
      pM = p75*(T75/TM)**(goR/gam1)                                     STRA 34
```

Interpolates for temperature Tz and pressure pz at height z, using linear temperature gradients in whichever of the two layers the height z falls (H75 to HM or HM to HFP)

```
      If (HP .lt. HM)Then                                               STRA 37
        tgrad = gam1                                                    STRA 38
        Tz = T75 + tgrad*(HP - H75)                                     STRA 39
        expon = goR/gam1                                                STRA 40
        pz = p75*(T75/Tz)**expon                                        STRA 41
      Else                                                              STRA 42
        tgrad = gam2                                                    STRA 43
        Tz = TM + tgrad*(HP - HM)                                       STRA 44
        expon = goR/gam2                                                STRA 45
        pz = pM*(TM/Tz)**expon                                          STRA 46
      Endif                                                             STRA 47
```

Convert the units of the temperature gradient to K/km

```
      tgrad = 0.001*tgrad                                               STRA 49
```

Computes the density dz at height z from the perfect gas law

```
  100 dz = M*pz/(Rstar*Tz)                                              STRA 51
```

Calculates the scale height Hz at height z by the relation

```
      Hz = 0.001*Rstar*Tz/(gz*M)                                        STRA 53
```

Subroutine: STEWART2 **Code:** STW2

Description: Stewart thermosphere model for computing global mean temperature and density at a given areocentric longitude of the sun (Ls). Computes effects due to solar activity influence and effects due to dust storm (if one is in progress).

Called By: ATMOS2

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

THERM

Input Variables (not passed through commons):

Name	Type	Description
ALS0. . . .	REAL*4	Ls value at start of dust storm (if any)
CHGT. . . .	REAL*4	current height
INTENS. . .	REAL*4	dust storm intensity (0.0 - 3.0)
IU0	INTEGER*4	unit number for screen output of messages
LAT	REAL*4	current latitude
LST	REAL*4	local solar time
LSUN. . . .	REAL*4	areocentric longitude of the sun (Ls)
RAUI. . . .	REAL*4	Mars orbital radius in astronomical units
RSTAR . . .	REAL*4	gas constant
SIGMA . . .	REAL*4	standard deviation for long term thermosphere variations

Output Variables (not passed through commons):

Name	Type	Description
H	REAL*4	pressure scale height
MOLWTG. . .	REAL*4	molecular weight
TOTALMDZ. .	REAL*4	total atmospheric mass density
TOTALPRZ. .	REAL*4	total atmospheric pressure
TZ.	REAL*4	atmospheric temperature

Local Variables (not passed through commons):

Name	Type	Description
DR.	REAL*4	seasonal correction to the height of the base of the thermosphere
DUST. . . .	REAL*4	dust correction to the height of the base of the thermosphere
ES.	REAL*4	array of thermospheric correction factors
FBAR. . . .	REAL*4	mean value of 10.7 cm solar flux at Earth position (1 AU)
FBARR . . .	REAL*4	mean value of 10.7 cm solar flux at Mars position
FLAG. . . .	INTEGER*4	test flag, set to 1 to get diagnostic output
GZ.	REAL*4	acceleration of gravity at current height
MDZ	REAL*4	atmospheric mass density
NDZ	REAL*4	atmospheric number density
PFAC. . . .	REAL*4	relative pressure factor for season and latitude
PRZ	REAL*4	atmospheric pressure
RAU	REAL*4	Mars orbital radius (in AU)
RF.	REAL*4	total radius to position (height plus ellipsoid radius)
RREF. . . .	REAL*4	local radius of reference ellipsoid
SMA	REAL*4	semi-major axis of Mars orbit
TF.	REAL*4	temperature at the base of the thermosphere
TINF. . . .	REAL*4	exospheric temperature
TO.	REAL*4	factor used in correction to ZF for seasonal variation

TOTALNDZ. .	REAL*4	total atmospheric number density
ZF.	REAL*4	height of the base of the thermosphere
ZZF	REAL*4	difference between current height and ZF

Methodology:

The Stewart thermosphere model (including subroutine STEWART2 and the associated subroutines THERMOS, DZDUST, Escalc, PRSEAS and RELLIPS) was converted to FORTRAN code from the Pascal version listed in Appendix B of "The Mars Atmosphere: Observations and Model Profiles for Mars Missions", David E. Pitts et al., eds., JSC-24455. The model has also been discussed, and given in IDL code, by Ian Stewart, Laboratory for Atmospheric and Space Physics, University of Colorado, Final Report JPL PO # NQ-802429.

Use the EScalc subroutine to evaluate the ES array of thermospheric variability factors for the input standard deviations for short term (stdl) and long term (SIGMA) variability

```
      Call EScalc(stdl,SIGMA,ES)                                  STW2 21
```

Adjust the 10.7 cm solar flux for long-term mean value, with ES(0)

```
      FBAR = F107 * EXP(ES(0))                                    STW2 25
```

Convert the 10.7 cm solar flux to the orbital position of Mars

```
      FBARR = FBAR / (RAU**2)                                     STW2 29
```

Call RELLIPS to get the acceleration of gravity, GZ, and the local radius of the reference ellipsoid, RREF. Call PRSEAS to get the relative pressure factor for the given value of Ls (LSUN) and latitude.

```
      CALL RELLIPS( LAT, RREF, CHGT, GZ)                          STW2 30
      CALL PRSEAS(LSUN, LAT, PFAC)                                STW2 31
```

Calculate DR, the seasonal correction to the height of the base of the thermosphere

```
      TO = 220.0 * SMA / RAU                                      STW2 33
      DR = (TO / 19.51) * ALOG(PFAC)                              STW2 34
```

Evaluate the dust storm correction and compute ZF, the height of the base of the thermosphere, and ZZF, the current height relative to the base of the thermosphere

```
      CALL DZDUST( LSUN, als0, INTENS, DUST)                      STW2 36
      DUST = DUST * EXP(ES(10))                                   STW2 37
      ZF = (124.4 * SMA / RAU) * EXP(ES(8) + ES(9)) + DR + DUST   STW2 38
      ZZF = CHGT - ZF                                             STW2 39
```

Find the total radius (radius of reference ellipsoid plus the height), the exospheric temperature, TINF, and the temperature of the base of the thermosphere, TF

```
      RF = RREF + ZF                                              STW2 45
      TINF = 4.11 * (11.0 + FBARR) * EXP(ES(2) + ES(3))           STW2 46
```

```
      TF = (170.0 * SMA / RAU) * EXP(ES(8) + ES(9))                        STW2 47
```

Call the THERMOS subroutine to get the remaining variables computed by the Stewart model

```
      CALL THERMOS(FLAG, ES, TINF, TF, LAT, LST, ZF,RF, ZZF, TOTALPRZ,     STW2 58
     & TOTALNDZ, TZ, MOLWTG, PRZ, NDZ, MDZ, TOTALMDZ ,iu0)                 STW2 59
```

Calculate H, the scale height, in km

```
      H = RSTAR*TZ/(1000.*MOLWTG*GZ)                                       STW2 61
```

Convert the atmospheric pressure value to N/m**2

```
      TOTALPRZ = TOTALPRZ*1.0E5                                            STW2 63
```

Convert the atmospheric density value to kg/m**3

```
      TOTALMDZ = TOTALMDZ*1000.                                            STW2 65
```

Function: Tdiurnal **Code:** TDIR

Description: Computes local surface temperature from the daily maximum and minimum surface temperature and the site-sun longitude separation.

Called By: Tsurface

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
DLO	REAL*4	longitude difference from current position to sun
TMAX. . . .	REAL*4	daily maximum surface temperature
TMIN. . . .	REAL*4	daily minimum surface temperature

Output Variables (not passed through commons):

Name	Type	Description
TDIURNAL. .	REAL*4	local surface temperature at current time

Local Variables (not passed through commons):

Name	Type	Description
COSFAC. . .	REAL*4	cosine factor in diurnal variation (+1 or -1)
DT.	REAL*4	diurnal temperature interpolation factor
OFFSET. . .	REAL*4	longitude offset in diurnal temperature interpolation
PERIOD. . .	REAL*4	effective period for diurnal interpolation
PI180 . . .	REAL*4	π / 180

Methodology:

Divides site-sun longitude difference into five ranges (-180 to -102.19 degrees, -102.19 to -48.6 degrees, -48.6 to +43.8 degrees, +43.8 to +111.5 degrees, and +111.5 to +180 degrees). Finds values of cosfac, period and offset for whichever range is appropriate, based on input value of site-sun longitude difference. Computes the diurnal temperature shape factor dT for the values of cosfac, period and offset thus determined

```
dT = .5*(1. + cosfac*cos(pi180*period*(dlo + offset)))          TDIR 26
dT = (-0.3455*dT + 1.3455)*dT                                   TDIR 27
```

Evaluates the local surface temperature, from the interpolation factor dT and the daily maximum and minimum temperature values

```
Tdiurnal = Tmin + (Tmax - Tmin)*dT                              TDIR 29
```

Function: Terrain **Code:** TERN

Description: Computes terrain height, relative to reference ellipsoid, for the given array of terrain height data, at a given latitude and longitude.

Called By: MarsGRAM (Main), Datastep, SETUP

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

TERHGT

Input Variables (not passed through commons):

Name	Type	Description
ALAT. . . .	REAL*4	current latitude (degrees)
ALON. . . .	REAL*4	current West longitude (degrees)

Output Variables (not passed through commons):

Name	Type	Description
TERRAIN . .	REAL*4	terrain height (km)

Local Variables (not passed through commons):

Name	Type	Description
DLAT. . . .	REAL*4	latitude increment between terrain data base values
DLON. . . .	REAL*4	longitude increment between terrain data base values
F1 thru F4	REAL*4	terrain heights at four corners of grid containing the desired location
ILAT. . . .	INTEGER*4	latitude index of terrain height data array
ILON. . . .	INTEGER*4	longitude index of terrain height data array
NLAT. . . .	INTEGER*4	number of latitude grids in data array
NLON. . . .	INTEGER*4	number of longitude grids in data array
RADVL1. . .	REAL*4	normalized distance parameter from Viking 1 site
RADVL2. . .	REAL*4	normalized distance parameter from Viking 2 site
TERR. . . .	REAL*4	intermediate value of interpolated terrain height
VL1HGT. . .	REAL*4	Viking 1 site terrain height
VL1LAT. . .	REAL*4	Viking 1 site latitude
VL1LON. . .	REAL*4	Viking 1 site longitude
VL2HGT. . .	REAL*4	Viking 2 site terrain height
VL2LAT. . .	REAL*4	Viking 2 site latitude
VL2LON. . .	REAL*4	Viking 2 site longitude
XLON. . . .	REAL*4	relative longitude from interpolation grid corner
YLAT. . . .	REAL*4	relative latitude from interpolation grid corner

Methodology:

Computes the latitude and longitude increments between the terrain data bases grid points

```
dlat = 180./(nlat-1.)                                          TERN 14
dlon = 360./(nlon-1.)                                          TERN 15
```

Finds the latitude and longitude index values of the interpolation lat-lon rectangle that contains the desired location

```
      ilat = 1 + int((90.+alat)/dlat)                                  TERN 17
      If (ilat .gt. nlat-1)ilat = nlat-1                               TERN 18

      ilon = 1 + int(alon/dlon)                                        TERN 20
      If (ilon .gt. nlon-1)ilon = nlon-1                               TERN 21
```

Gets the terrain heights at the corner points of the interpolation rectangle from the terrain height data base array

```
      F1 = th(ilat,ilon)                                               TERN 23
      F2 = th(ilat,ilon+1)                                             TERN 24
      F3 = th(ilat+1,ilon)                                             TERN 25
      F4 = th(ilat+1,ilon+1)                                           TERN 26
```

Computes normalized latitude and longitude displacements from the corner of the interpolation rectangle

```
      ylat = 1. + (90.+alat)/dlat - ilat                               TERN 28

      xlon = 1. + (alon/dlon) - ilon                                   TERN 30
```

Does a bi-linear interpolation across the interpolation rectangle to find the terrain height at the desired location

```
      Terr = F1 + (F2 - F1)*xlon + (F3 - F1)*ylat                      TERN 32
     & + (F4 - F2 - F3 + F1)*xlon*ylat                                 TERN 33
```

Finds the normalized distances from the Viking 1 and Viking 2 sites

```
      radvl1 = ((alat-vl1lat)**2 + (alon-vl1lon)**2)/4.0               TERN 35
      radvl2 = ((alat-vl2lat)**2 + (alon-vl2lon)**2)/4.0               TERN 36
```

Modifies the interpolated terrain value so that there is a smooth transition to the Viking site value, if the location is within about 1 degree of lat-lon from either of the Viking sites

```
      If (radvl1 .lt. 1.0)then                                         TERN 38
        Terr = (1.-radvl1)*vl1hgt + Terr*radvl1                        TERN 39
      Else If(radvl2 .lt. 1.0)then                                     TERN 40
        Terr = (1.-radvl2)*vl2hgt + Terr*radvl2                        TERN 41
      Endif                                                            TERN 42
      Terrain = Terr                                                   TERN 43
```

Subroutine: THERMOS **Code:** THRM

Description: Subroutine for the Stewart thermosphere model. Returns the temperature, pressure, molecular weight, mass density and number density concentrations versus altitude above the base of the thermosphere.

Called By: STEWART2

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
ES.	REAL*4	array of thermospheric variability factors for the input standard deviations for short term (stdl) and long term (SIGMA) variability
FLAG. . . .	REAL*4	test flag, set to 1 to get diagnostic output
IU0	INTEGER*4	unit number for screen output of messages
LAT	REAL*4	current latitude
LST	REAL*4	local solar time
RF.	REAL*4	total radius (radius of reference ellipsoid plus height)
TF.	REAL*4	temperature at base of thermosphere
TINF. . . .	REAL*4	exospheric temperature
ZF.	REAL*4	height of base of thermosphere
ZZF	REAL*4	current height above height ZF

Output Variables (not passed through commons):

Name	Type	Description
MDZ	REAL*4	atmospheric constituent mass density
MOLWTG. . .	REAL*4	molecular weight
NDZ	REAL*4	atmospheric constituent number density
PRZ	REAL*4	atmospheric constituent partial pressure
TOTALMDZ. .	REAL*4	total atmospheric mass density
TOTALNDZ. .	REAL*4	total atmospheric number density
TOTALPRZ. .	REAL*4	total atmospheric pressure
TZ.	REAL*4	local temperature at current height

Local Variables (not passed through commons):

Name	Type	Description
AO.	REAL*4	factor in atomic oxygen calculation
BK.	REAL*4	numerical constant for the number density calculations
DM.	REAL*4	molecular mass array
FF.	REAL*4	fractional composition of the heterosphere
FO.	REAL*4	factor in atomic oxygen calculation
GF.	REAL*4	acceleration of gravity at height ZF
HH.	REAL*4	scale heights of the heavy gas constituents
I	INTEGER*4	index for heavy gas calculations
J	INTEGER*4	index for light gas calculations
K	INTEGER*4	dummy index in implied loop for write statement
M	REAL*4	molecular weights for the constituent gases
P1BAR . . .	REAL*4	pressure of 1 bar
PFH	REAL*4	partial pressure of H at height ZF
PFH2. . . .	REAL*4	partial pressure of H2 at height ZF
PFHE. . . .	REAL*4	partial pressure of He at height ZF
PRESSF. . .	REAL*4	pressure at height ZF

RADEG . . .	REAL*4	180 / π
RATIO . . .	REAL*4	factor used in computing partial pressure of H
RREF. . . .	REAL*4	local radius of reference ellipsoid
SCALE . . .	REAL*4	scale factor used in temperature calculation
XDM	REAL*4	molecular mass array for light gases
XFF	REAL*4	partial pressure array for He, H2 and H
XHH	REAL*4	scale heights of the light gas constituents
YSC	REAL*4	scaled height parameter in the temperature calculation

Methodology:

The Stewart thermosphere model (including the main subroutine STEWART2 and the associated subroutines THERMOS, DZDUST, Escalc, PRSEAS and RELLIPS) was converted to FORTRAN code from the Pascal version listed in Appendix B of "The Mars Atmosphere: Observations and Model Profiles for Mars Missions", David E. Pitts et al., eds., JSC-24455. The model has also been discussed, and given in IDL code, by Ian Stewart, Laboratory for Atmospheric and Space Physics, University of Colorado, Final Report JPL PO # NQ-802429.

Calls RELLIPS to get local radius of reference ellipsoid, RREF, and GF, the acceleration of gravity at the height ZF, the base of the thermosphere

```
      CALL RELLIPS(LAT,RREF,ZF,GF)                                THRM 27
```

Sets the pressure at the base of the thermosphere to 1.26 nanobars (1.26×10^{-4} N/m^2)

```
      PRESSF = 1.26E-9                                            THRM 31
```

Sets P1BAR to the pressure of 1 bar (1.0×10^5 N/m^2)

```
      P1BAR = 1.0E6                                               THRM 34
```

Computes A0 and F0, parameters for use in calculating the atomic oxygen concentration

```
      AO = 0.18 * (1.0 + ES(7))                                   THRM 36

      FO = 0.01 * EXP(ES(4) + ES(5))                              THRM 38
```

Evaluates SCALE, the scale factor used in the temperature versus height

```
      SCALE = TF / 9.20                                           THRM 40
```

Sets the molecular weight constant arrays M, DM and XDM (THRM 41 through THRM 76). Sets the array FF, for the composition of the heterosphere (THRM 77 through THRM 89). Adjusts FF(5) for atomic oxygen

```
      FF(5) = FO*(1.0-AO*SIN(15.0*LST/RADEG)*COS(LAT/RADEG))      THRM 90
```

Computes exobase (height ZF) partial pressures for helium (PFHE), molecular hydrogen (PFH2) and atomic hydrogen (PFH) and converts these to partial pressures (XFF array)

```
      PFHE = 3.3E-16 * TINF                                       THRM 91
```

```
      PFH2 = 2.4E-15

      if (TINF .LE. 330.0)PFH=5.2E-16*TINF*EXP(-TINF/70.0)

      if (TINF .GT. 330.0) then
        RATIO = 1440.0 / TINF
        PFH=5.8E-18*SQRT(TINF)*EXP(RATIO)/(1.0+RATIO)

      ENDIF
      XFF(0) = PFHE / PRESSF
      XFF(1) = PFH2 / PRESSF
      XFF(2) = PFH / PRESSF
```

Calculates local atmospheric temperature at current height

```
      YSC = ZZF * RF / (RF + ZZF)
      TZ = TINF - (TINF - TF) * EXP(-YSC / SCALE)
```

Loops through the heavy gases (I = 0 for CO2, 1 for N2, 2 for Argon, 3 for O2, 4 for CO, and 5 for atomic oxygen), and computes scale height, HH(I), partial pressure, PRZ(I), number density, NDZ(I), mass density, MDZ(I), and accumulates total mass density, TOTALMDZ, total pressure, TOTALPRZ, and pressure-weighted molecular weight, MOLWTG.

```
      DO 200 I = 0, 5
        HH(I) = BK * TINF / (GF * DM(I)) / 1.0E5

        PRZ(I) = PRESSF*FF(I)*EXP(-YSC/HH(I)-(SCALE/HH(I))*ALOG(TZ/TF))
        NDZ(I) = P1BAR * PRZ(I) / (BK * TZ)

        MDZ(I) = NDZ(I) * DM(I)
        TOTALMDZ = TOTALMDZ + MDZ(I)
        TOTALPRZ = TOTALPRZ + PRZ(I)
        MOLWTG = MOLWTG + PRZ(I) * M(I)
  200 CONTINUE
```

Loops through the light gases (J = 0 for He, 1 for H2, 2 for H), and computes scale height, XHH(J), partial pressure, PRZ(J+6), number density, NDZ(J+6), mass density, MDZ(J+6), and accumulates total mass density, TOTALMDZ, total pressure, TOTALPRZ, and pressure-weighted molecular weight, MOLWTG.

```
      DO 210 J = 0, 2
        XHH(J) = BK * TINF / (GF * XDM(J)) / 1.0E5

        PRZ(J+6)=PRESSF*XFF(J)*EXP(-YSC/XHH(J)-(SCALE/XHH(J))*
     &   ALOG(TZ/TF))
        NDZ(J+6) = P1BAR * PRZ(J + 6) / (BK * TZ)

        MDZ(J + 6) = NDZ(J + 6) * XDM(J)
        TOTALMDZ = TOTALMDZ + MDZ(J + 6)
        TOTALPRZ = TOTALPRZ + PRZ(J + 6)
        MOLWTG = MOLWTG + PRZ(J + 6) * M(J + 6)
  210 CONTINUE
```

Divides by total pressure to get true total molecular weight, and computes total number density, TOTALNDZ

```
      MOLWTG = MOLWTG / TOTALPRZ
      TOTALNDZ = P1BAR * TOTALPRZ / (BK * TZ)
```

Subroutine: Temps **Code:** TMPS

Description: Computes array of atmospheric temperatures at the significant levels for a given latitude and longitude. Dust effects are included.

Called By: ATMOS2

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
ALAT. . . .	REAL*4	local site latitude
ALS	REAL*4	areocentric longitude of the sun (Ls)
DLON. . . .	REAL*4	site-sun West longitude difference
DUSTA . . .	REAL*4	magnitude of dust storm effect on daily amplitude
DUSTM . . .	REAL*4	magnitude of dust storm effect on daily average
GAM	REAL*4	lapse rates (-dT/dz) between significant levels
T0BAR . . .	REAL*4	average surface temperature with no dust storm effect
TSURF . . .	REAL*4	local surface temperature, including dust effect

Output Variables (not passed through commons):

Name	Type	Description
T	REAL*4	array of air temperatures at the significant levels
GAM	REAL*4	lapse rates, modified by dust storm effects

Local Variables (not passed through commons):

Name	Type	Description
A25	REAL*4	coefficient array in calculation of T25
ABSLAT. . .	REAL*4	absolute value of latitude
AFACT . . .	REAL*4	factor array for diurnal temperature variation
ALATP . . .	REAL*4	90 minus absolute latitude
ALONMAX . .	REAL*4	longitude where maximum phase occurs
AT25. . . .	REAL*4	latitude dependent coefficient in T25 calculation
B25	REAL*4	coefficient array in calculation of T25
BT25. . . .	REAL*4	latitude dependent coefficient in T25 calculation
C25	REAL*4	coefficient array in calculation of T25
COSLAT. . .	REAL*4	cosine latitude term in T25 calculation
CT25. . . .	REAL*4	latitude dependent coefficient in T25 calculation
DFACTOR . .	REAL*4	factor array for dust storm effect on diurnal amplitude
DT25. . . .	REAL*4	difference between T25 and value from lapse rates
DTR26 . . .	REAL*4	diurnal amplitude factor for 26 km
DTR30 . . .	REAL*4	diurnal amplitude factor for 30 km
DZ.	REAL*4	array of height differences between significant levels
FACTOR. . .	REAL*4	height scaling factors for diurnal amplitude (relative to amplitude at 30 km) in non-dust storm conditions
I	INTEGER*4	array index
PI120 . . .	REAL*4	π / 120
PI180 . . .	REAL*4	π / 180
SINLAT. . .	REAL*4	sine latitude term in T25 calculation
T25	REAL*4	air temperature at 25 km
THOUR . . .	REAL*4	phase hour of maximum diurnal perturbation
THOUR0. . .	REAL*4	phase hour for maximum diurnal perturbation at surface
THOUR26 . .	REAL*4	phase hour for maximum at 26 km height
Z	REAL*4	height array for significant levels

Methodology:

Starts by computing an estimate of the temperature at 25 km, T25, from a set of coefficients for latitude and Ls dependence

```
      AT25 = A25(0)                                                TMPS 41
      BT25 = B25(0)                                                TMPS 42
      CT25 = C25(0)                                                TMPS 43
      Do 5 i = 1,5,2                                               TMPS 44
        sinlat = sin(pi120*(i+1.)*alat/2.)                         TMPS 45
        coslat = cos(pi120*(i+1.)*alat/2.)                         TMPS 46
        AT25 = AT25 + A25(i)*sinlat                                TMPS 47
        AT25 = AT25 + A25(i+1)*coslat                              TMPS 48
        BT25 = BT25 + B25(i)*sinlat                                TMPS 49
        BT25 = BT25 + B25(i+1)*coslat                              TMPS 50
        CT25 = CT25 + C25(i)*sinlat                                TMPS 51
    5 CT25 = CT25 + C25(i+1)*coslat                                TMPS 52
      T25 = AT25 + BT25*sin(pi180*als) + CT25*cos(pi180*als)       TMPS 53
```

Computes the daily average atmospheric temperature (no dust storm correction) at each of the significant levels, from the daily average surface temperature and the lapse rates between the significant levels

```
      T(0) = T0bar                                                 TMPS 56
      Do 10 i = 1,3                                                TMPS 57
   10 T(i) = T(i-1) - gam(i)*dz(i)                                 TMPS 58
```

Finds the difference from the temperature estimate T25 and the temperature at 25 km, computed from the lapse rates

```
      DT25 = T25 - (T(2) + 2.*T(3))/3.                             TMPS 60
```

Adjusts the temperature and lapse rates for the first three significant levels, based on the value of T25

```
      Do 15 i = 1,3                                                TMPS 62
        T(i) = T(i) + z(i)*DT25/25.                                TMPS 63
   15 gam(i) = (T(i-1) - T(i))/dz(i)                               TMPS 64
```

Adjusts the temperatures at the 4th and 5th significant levels, based on the new value of T(3)

```
      Do 20 i = 4,5                                                TMPS 66
   20 T(i) = T(i-1) - gam(i)*dz(i)                                 TMPS 67
```

Compute non-dust storm effects if dustM = 0 (TMPS 70 - TMPS 81). Calculates factor = afact, the latitude dependence factor for the diurnal temperature variation at 5 km height

```
      If (dustM .eq. 0)then                                        TMPS 70

        If (abslat .le. 25)then                                    TMPS 72
          Factor(1) = 0.79                                         TMPS 73
        Else If (abslat .ge. 45)then                               TMPS 74
          Factor(1) = 0.43                                         TMPS 75
```

```
        Else                                                TMPS 76
          Factor(1) = 1.24 - 0.018*abslat                   TMPS 77
        Endif                                               TMPS 78
        Do 25 i = 1,5                                       TMPS 79
   25   afact(i) = factor(i)                                TMPS 80
      Endif                                                 TMPS 81
```

Computes the relative temperature amplitude at 30 km height

```
        dtr30 = 0.0141*(90. - abslat)**0.2                  TMPS 84
```

Evaluates thour26, the phase hour for maximum temperature at 26 km height

```
      If (alat .le. -70.)then                               TMPS 89
        thour26 = 21.                                       TMPS 90
      Else If (alat .ge. -20.)then                          TMPS 91
        thour26 = 11.5                                      TMPS 92
      Else                                                  TMPS 93
        thour26 = 7.7 - 0.19*alat                           TMPS 94
      Endif                                                 TMPS 95
```

Computes the effects of a dust storm on the daily average temperature and on the diurnal temperature variation, if a dust storm is in progress (TMPS 96 - TMPS123).

```
      If (dustM .gt. 0.0)then                               TMPS 96

C...    Adjust hour of diurnal temperature maximum at 26 km TMPS 98
        thour26 = (1. - dustA)*thour26 + dustA*16.7         TMPS 99
C...    Adjust height factors for diurnal amplitudes        TMPS100
        Do 30 i = 1,5                                       TMPS101
   30   afact(i) = (1. - dustA)*factor(i) + dustA*dfactor(i) TMPS102
```

Adjustments for dust storm effects on the daily average temperature for latitudes less than 50 degrees (absolute value) are done in TMPS104 through TMPS110. For absolute latitude greater than 50 degrees, the adjustments are done in TMPS111 through 120.

```
        If (abslat .le. 50.)then                            TMPS104
          dtr26 = 0.0465 + 1.90E-6*abslat**2                TMPS105
          T(1) = T(1) + 13.*dustM                           TMPS106
          T(2) = T(2) + 36.*dustM                           TMPS107
          T(3) = T(3) + (48. + 0.0024*abslat**2)*dustM      TMPS108
          T(4) = T(4) + 30.*dustM                           TMPS109
          T(5) = T(5) + 26.*dustM                           TMPS110
        Else                                                TMPS111
          alatp = 90. - abslat                              TMPS112
          dtr26 = ((1.597E-6*alatp - 1.417E-4)*alatp +      TMPS113
     &      4.394E-3)*alatp                                 TMPS114
          T(1) = T(1) + (28. - 0.3*abslat)*dustM            TMPS115
          T(2) = T(2) + (11. + 0.015625*alatp**2)*dustM     TMPS116
          T(3) = T(3) + (25. + 0.018125*alatp**2)*dustM     TMPS117
          T(4) = T(4) + (50. - 0.4*abslat)*dustM            TMPS118
          T(5) = T(5) + (46. - 0.4*abslat)*dustM            TMPS119
        Endif                                               TMPS120
```

Adjusts the diurnal amplitude from 26 to 30 km

```
        dtr30 = (1. - dustA)*dtr30 + dustA*1.072*dtr26
      Endif
```

Steps through heights of the significant levels and evaluates the temperatures, T(i), including dust storm effects and diurnal variations

```
      Do 40 i = 1,5
C...    Compute diurnal adjustment to average temperatures.
C...    Interpolate (extrapolate) for phase hour of maximum diurnal
C       perturbation
        thour = thour0 + (thour26 - thour0)*z(i)/26.
C...    Make sure phase hour is in range 0 < thour < 24
        If (thour .gt. 24.)thour = thour - 24.
        If (thour .lt. 0.)thour = thour + 24.
C...    Longitude where maximum phase occurs
        alonmax = 15.*(thour - 12.)
C...    Evaluate temperature including diurnal perturbation
        T(i) = T(i)*(1. + dtr30*afact(i)*cos(pi180*(dlon + alonmax)))
   40 Continue
      T(0) = Tsurf
```

Re-evaluates the lapse rates (K/km), based on the new values of temperatures at the significant levels

```
      Do 50 i = 1,5
   50 gam(i) = (T(i-1) - T(i))/dz(i)
```

Subroutine: Tsurface **Code:** TSRF

Description: Computes surface temperature for a given latitude and longitude, areocentric longitude of the sun (Ls) and orbital radius. Also computes the diurnal average surface temperature, with and without dust storm effects.

Called By: ATMOS2

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

None

Input Variables (not passed through commons):

Name	Type	Description
ALS. . . .	REAL*4	areocentric longitude of the sun (Ls)
AU.	REAL*4	orbital radius of Mars from Sun (in astronomical units)
DUSTT . . .	REAL*4	relative dust storm effect (0-1)
SITLA . . .	REAL*4	current latitude
SITLO . . .	REAL*4	current longitude
SUNLA . . .	REAL*4	latitude of sub-solar point on surface
SUNLO . . .	REAL*4	West longitude of sub-solar point on surface

Output Variables (not passed through commons):

Name	Type	Description
T0BAR . . .	REAL*4	daily average surface temperature without dust effects
TAVG. . . .	REAL*4	daily average surface temperature including dust effect
TMAX. . . .	REAL*4	daily maximum surface temperature
TMIN. . . .	REAL*4	daily minimum surface temperature
TSURF . . .	REAL*4	local surface temperature at current position and time

Local Variables (not passed through commons):

Name	Type	Description
ABSLAT. . .	REAL*4	absolute value of latitude
ABSORB. . .	REAL*4	daily total solar radiation absorbed at surface
ALBEDO. . .	REAL*4	surface reflectance for solar radiation
ATRANS. . .	REAL*4	factor used to compute average solar transmittance
AVCOS . . .	REAL*4	daily average value of the cosine of the solar zenith angle at the site latitude
AVMU. . . .	REAL*4	cosine of the solar zenith angle at solar noon
CAPNORTH. .	REAL*4	boundary of the north polar cap
CAPSOUTH. .	REAL*4	boundary of the south polar cap
CLAT. . . .	REAL*4	cosine of the current latitude
CLATS . . .	REAL*4	cosine of the solar latitude
COSDLON . .	REAL*4	cosine of the hour-angle length of the solar day
CRLAT . . .	REAL*4	latitude of the boundary at which the daily total amount of solar radiation is 0 (total darkness for the day)
DELTA . . .	REAL*4	optical depth for a vertical path through the atmosphere
DLO	REAL*4	longitude difference from sun to site (also equal to the solar hour angle in degrees, measured from solar noon)
DLON. . . .	REAL*4	hour angle length of the solar day at the site position
DTR	REAL*4	π / 180
F0.	REAL*4	the solar constant at current Mars orbital radius
PI.	REAL*4	π
PMMU. . . .	REAL*4	minimum value of the cosine of the solar zenith angle

POLAR . . .	REAL*4	factor for temperature correction in the polar cap zone
QA.	REAL*4	amplitude of the daily variation in solar heating at the surface (difference between average and minimum solar heating)
SINDLON . .	REAL*4	sine of the hour-angle length of the solar day
SITELAT . .	REAL*4	current latitude in radians
SLAT. . . .	REAL*4	sine of current latitude
SLATS . . .	REAL*4	sine of the solar latitude
SSALB . . .	REAL*4	single scatter albedo for atmospheric scattering
SUNLAT. . .	REAL*4	solar latitude in radians
TAMP. . . .	REAL*4	amplitude of the diurnal surface temperature variation
TAUBAR. . .	REAL*4	daily average of the solar transmittance to the surface
TLAT. . . .	REAL*4	tangent of the current latitude
TLATS . . .	REAL*4	tangent of the solar latitude

Methodology:

Computes cosine of the hour-angle length of the solar day at the site latitude

```
      cosdlon = -tlat*tlats                                          TSRF 28
```

Calculates crlat, the latitude at which daily insolation = 0, dlon, the length of the solar day, and polar, the temperature adjustment factor in the polar cap regions at locations where the daily insolation is zero. If |cosdlon| > 1 then either the insolation lasts all day (sun never sets, dlon = π radians) or there is no daylight (sun never rises, dlon = 0).

```
      If (Abs(cosdlon) .gt. 1.)then                                  TSRF 32
        If(sitla*sunla.lt.0.)Then                                    TSRF 32a

          If (sunla.gt.0.0)Then                                      TSRF 33a
            crlat = sunla - 90.                                      TSRF 33b
          Else                                                       TSRF 33c
            crlat = sunla + 90.                                      TSRF 33d
          Endif                                                      TSRF 33e

          polar = (abs(sitla-crlat))/25.19                           TSRF 35
          dlon = 0.                                                  TSRF 36
        Else                                                         TSRF 37
          dlon = pi                                                  TSRF 38
        Endif                                                        TSRF 39
        sindlon = 0.                                                 TSRF 40
      Else                                                           TSRF 44
        sindlon = sqrt(1. - cosdlon**2)                              TSRF 45
        dlon = pi/2.                                                 TSRF 46
        If(cosdlon.ne.0.)dlon = atan(sindlon/cosdlon)                TSRF 47
        If (dlon .le. 0.0)dlon = pi + dlon                           TSRF 48
      Endif                                                          TSRF 48a
```

Computes avcos, the daily average of the cosine of the solar zenith angle at the site latitude

```
      avcos = (slats*slat*dlon + clats*clat*sindlon)/pi              TSRF 50
      If (avcos .lt. 0.0)avcos = 0.                                  TSRF 51
```

Sets delta, the dust optical depth and ssalb, the single-scatter albedo to model values, and computes avmu, the cosine of the solar zenith angle at solar noon.

```
      delta = 0.3                                                  TSRF 51b
      ssalb = 0.85                                                 TSRF 51c
      atrans = ssalb/2.                                            TSRF 51d

      avmu = slat*slats + clat*clats                               TSRF 51f
```

Computes taubar, the daily-average solar transmittance

```
      If (avmu.le.0.0)Then                                         TSRF 51h
        avmu = 0.0                                                 TSRF 51i
        taubar = 0.0                                               TSRF 51j
      Else                                                         TSRF 51k
        taubar = atrans + (1. - atrans)*exp(-delta*(1./avmu))      TSRF 51l
      Endif                                                        TSRF 51m
```

Uses the Alb function to get the surface albedo

```
      albedo = Alb(als,sitla)                                      TSRF 51o
```

Calculates the daily total amount of solar radiation absorbed at the surface

```
      Absorb = taubar*(1. - albedo)*F0*avcos                       TSRF 53
```

Finds pmmu, the cosine of the minimum of the solar zenith angle

```
      pmmu = slat*slats - clat*clats                               TSRF 53b
      If (pmmu.lt.0.0)pmmu = 0.0                                   TSRF 53c
```

Evaluates Qa, the amplitude of the diurnal surface heating

```
      Qa = taubar*(1. - albedo)*F0*(avmu - pmmu)/2.                TSRF 53e
```

Determines the limits of the northern and southern polar caps

```
      capnorth = 90. - polecap(90.,als)                            TSRF 53g
      capsouth = -90. + polecap(-90.,als)                          TSRF 53h
```

Computes the daily average surface temperature, including the polar correction if in the polar cap region

```
      If (sitla.gt.capnorth.or.sitla.lt.capsouth)Then              TSRF 53j
        Tavg = 140.5 + 0.2336*Absorb - 8.5*polar                   TSRF 53k
      Else                                                         TSRF 53l
        Tavg = 140.5 + 0.8221*Absorb - 0.001425*(Absorb**2)        TSRF 53m
      Endif                                                        TSRF 53n
```

Calculates Tamp, the amplitude of the daily surface temperature variation

```
      Tamp = 0.16*Qa                                               TSRF 53p
```

Finds the minimum and maximum daily temperatures at site latitude

```
      Tmax = Tavg + Tamp                                           TSRF 55
      Tmin = Tavg - Tamp                                           TSRF 56
```

Saves the daily average surface temperature, without dust storm effect, as T0bar

```
      T0bar = Tavg                                                 TSRF 59
```

Compute the effects of a dust storm (if any) on Tavg, Tmax and Tmin

```
      If (dustT .gt. 0.0)then                                      TSRF 61
        abslat = abs(sitla)                                        TSRF 62
        If (abslat .le. 50.) then                                  TSRF 63
          Tavg = Tavg - (7. - 1.28E-6*abslat**4)*dustT             TSRF 64
        Else                                                       TSRF 65
          Tavg = Tavg + (7.5 - 0.13*abslat)*dustT                  TSRF 66
        Endif                                                      TSRF 67
        Tamp = (0.5 - 0.25*dustT)*(Tmax - Tmin)/T0bar              TSRF 68
        Tmax = Tavg*(1. + Tamp)                                    TSRF 69
        Tmin = Tavg*(1. - Tamp)                                    TSRF 70
      Endif                                                        TSRF 71
```

Finds the longitude difference from sun to site (equal solar hour angle, in degrees, measured from solar noon), and evaluates the local surface temperature from the minimum and maximum daily values

```
      dlo = sunlo - sitlo                                          TSRF 74
      if (dlo .gt. 180.)dlo = dlo - 360.                           TSRF 75
      if (dlo .lt. -180.)dlo = dlo + 360.                          TSRF 76
      Tsurf = Tdiurnal(dlo,Tmin,Tmax)                              TSRF 77
```

Subroutine: Wavepert **Code:** WAVE

Description: Zurek wave perturbations at given height, latitude and local time. Includes dust storm effects. Computes density perturbations from the original Zurek temperature wave perturbation.

Called By: Datastep

Common Blocks Used (See Table 2-3 for a list of variables in common blocks):

WAVEDAT

Input Variables (not passed through commons):

Name	Type	Description
CHGT. . . .	REAL*4	current height (km)
CLAT. . . .	REAL*4	current latitude (degrees)
DUSTA . . .	REAL*4	relative magnitude of dust storm effect on phase (0-1)
DUSTM . . .	REAL*4	relative magnitude of dust storm effect on amplitude (0-1)
TLOCAL. . .	REAL*4	local time in Mars hours (1/24th Sols)

Output Variables (not passed through commons):

Name	Type	Description
AMP1. . . .	REAL*4	diurnal amplitude of wave (% of mean)
AMP2. . . .	REAL*4	semi-diurnal amplitude of wave (% of mean)
WAVE. . . .	REAL*4	relative density perturbation of wave (% of mean)

Local Variables (not passed through commons):

Name	Type	Description
AMPC1 . . .	REAL*4	clear-atmosphere diurnal amplitude (% of mean)
AMPD1 . . .	REAL*4	dusty-atmosphere diurnal amplitude (% of mean)
AMPD2 . . .	REAL*4	dusty-atmosphere semi-diurnal amplitude (% of mean)
AMPI1 . . .	REAL*4	intermediate amplitude variable used in calculations
AMPI2 . . .	REAL*4	intermediate amplitude variable used in calculations
DLAT. . . .	REAL*4	relative latitude displacement from interpolation point
DZ.	REAL*4	intermediate value of relative height displacement
DZA	REAL*4	relative height displacement from interpolation point
H	REAL*4	height scale for vertical interpolation
I1.	INTEGER*4	1st height index for interpolation
I2.	INTEGER*4	2nd height index for interpolation
OMEGA1. . .	REAL*4	diurnal frequency (15 ° / Mars hour)
OMT1. . . .	REAL*4	phase-adjusted diurnal frequency
OMT2. . . .	REAL*4	phase-adjusted semi-diurnal frequency
PH1	REAL*4	diurnal phase (hours)
PH2	REAL*4	semi-diurnal phase (hours)
PHC1. . . .	REAL*4	clear-atmosphere diurnal phase (hours)
PHD1. . . .	REAL*4	dusty-atmosphere diurnal phase (hours)
PHD2. . . .	REAL*4	dusty-atmosphere semi-diurnal phase (hours)
PHI1. . . .	REAL*4	intermediate phase variable used in calculations
PHI2. . . .	REAL*4	intermediate phase variable used in calculations
Z1.	REAL*4	lower height for vertical interpolation
Z2.	REAL*4	upper height for vertical interpolation

Methodology:

The Zurek wave perturbation model was adapted from Section 4.3 of "The Mars Atmosphere: Observations and Model Profiles for Mars Missions", Report No. JSC-24455, David E. Pitts et al., eds., and the draft report by Pitts, Tillman, Pollack and Zurek, "Model Profiles of the Mars Atmosphere for the Mars Rover and Sample Return Mission", draft report, March 11, 1988.

Computes diurnal frequency (15 ° / Mars hour)

```
      omega1 = Atan(1.)/3.                                              WAVE 13
```

Evaluate i1 and i2, the array index values for height interpolation, the corresponding altitudes, z1 and z2, and dza, the relative height displacement between z1 and z2

```
      H = 11.26                                                         WAVE 14
      i1 = 1 + Int(2.*CHGT/H)                                           WAVE 15
      If (i1 .lt. 1)i1 = 1                                              WAVE 16
      If (i1 .gt. 11)i1 = 11                                            WAVE 17
      i2 = i1 + 1                                                       WAVE 18
      z1 = H*(i1 - 1.)/2.                                               WAVE 19
      z2 = H*i1/2.                                                      WAVE 20
      dz = 2.*(CHGT - z1)/H                                             WAVE 21
      dza = dz                                                          WAVE 22
      If (dz .gt. 1.)dza = 1.                                           WAVE 23
```

Uses the amplitude interpolation function (ampint) to interpolate on latitude (in the absolute latitude ranges 0-20°, 20-45° or > 45°). Uses the phase interpolation function (phasint) to interpolate the phase. Also uses ampint and phasint to do height interpolation . In the interpolation section (WAVE 24 - WAVE 77), the amplitude variables are designated "amp" and the phase variables are designated "ph", "c" denotes clear-atmosphere and "d" denotes dusty-atmosphere, 100, 120 and 145 denote diurnal variables at latitudes 0, 20 and 45°, respectively; and 200, 220 and 245 denote semi-diurnal variables at the same respective latitudes.

Use the ampint function to interpolate between clear and dusty diurnal amplitudes, based on the value of DustM, the dust magnitude effect. Also adjust the semi-diurnal amplitude (applicable only to the dusty case).

```
      amp1 = ampint(ampc1,ampd1,DustM)/100.                             WAVE 79
      amp2 = DustM*ampd2/100.                                           WAVE 80
```

Use the phasint function to interpolate between clear and dusty diurnal phases, based on the value of DustA, the magnitude of the dust effect on phase.

```
      ph1 = phasint(phc1,phd2,DustA,omega1)                             WAVE 81
```

Compute the phase-adjusted frequencies for the diurnal and semi-diurnal components, and evaluate the density wave perturbation from the temperature wave model values [see equations (1) through (4) in the section "The Zurek Wave Perturbation Model" of the Release 2 Technical Report, 1993; see Appendix C].

```
      omt1 = omega1*(TLOCAL - ph1)                                      WAVE 83
      omt2 = 2.*omega1*(TLOCAL - ph2)                                   WAVE 84
      wave = amp1*(sin(omt1) - cos(omt1)) + amp2*(sin(omt2) -           WAVE 85
     & cos(omt2))                                                       WAVE 86
```

APPENDIX B

MARS-GRAM RELEASE #1 TECHNICAL REPORT

This Appendix contains the technical portions of the Release #1 Report for Mars-GRAM (version 2.21), "The Mars Global Reference Atmospheric Model (Mars-GRAM)", Dale L. Johnson and Bonnie F. James (Grant Monitors), C. G. Justus and George Chimonas, October 8, 1989, prepared under Georgia Tech grant No. NAG8-078, for NASA Marshall Space Flight Center. Appendix material (giving outdated information on program input, output and running characteristics) has been deleted.

ABSTRACT

An engineering model atmosphere for Mars has been developed with many of the same features and capabilities of the highly successful Global Reference Atmospheric Model (GRAM) program for Earth's atmosphere, including mean values for density, temperature, pressure, and wind components, and density perturbation magnitudes and random perturbation profiles for density variations along specified trajectories. In the lower atmosphere of Mars (up to 75 km) the model is built around parameterizations of height, latitudinal, longitudinal and seasonal variations of temperature determined from a survey of published measurements from the Mariner and Viking programs. Pressure and density are inferred from the temperature by making use of the hydrostatic and perfect gas law relationships. For the upper atmosphere (above about 120 km), the thermospheric model of Stewart (1987) is used. A hydrostatic interpolation routine is used to insure a smooth transition from the lower portion of the model to the Stewart thermospheric model. Mars-GRAM includes parameterizations to simulate the effects of seasonal variation, diurnal variation, dust storm effects, effects due to the orbital position of Mars, effects of the large seasonal variation in surface atmospheric pressure because of differential condensation/ sublimation of the CO_2 atmosphere in the polar caps, and effects of Martian atmospheric mountain wave perturbations on the magnitude of the expected density perturbations. The thermospheric model includes a parameterization for the effects of solar activity, measured by the 10.7 cm solar radio flux. Winds are computed by an aereostrophic (thermal wind) approximation, with the inclusion of the effects of molecular viscosity, which, because of the low atmospheric densities, can be very important at high altitudes. The mountain wave perturbation model also includes a new damping approximation due to the effects of molecular viscosity.

INTRODUCTION AND BACKGROUND

A highly successful and well-utilized engineering model for the Earth's atmosphere, the Global Reference Atmospheric Model (GRAM), was developed at Georgia Tech (Justus, et al., 1975, 1976), and has undergone several improvement cycles (Justus and Roper 1987; Justus, 1988). GRAM applications include orbital mechanics and lifetime studies, vehicle design and performance criteria, attitude control analysis problems, analysis of effects of short-term density variation from geomagnetic storms, and aerobraking analyses (for missions requiring return from geosynchronous orbit to space-station rendezvous).

In addition to evaluating the mean density, temperature, pressure, and wind components at any height, latitude, longitude and monthly period, GRAM also allows for the simulation of "random perturbation" profiles about the mean conditions. This feature permits the simulation of a large number of realistic density profile realizations along the same trajectory through the atmosphere, with realistic values of scales of variation and peak perturbation values (e.g., the random perturbation profiles produce values which exceed the +3 standard deviation value approximately 1% of the time).

With the planning activity for upcoming and proposed unmanned missions to Mars (e.g. Mars Observer, Mars Aeronomy Observer, Mars Rover and Sample Return), as precursors to a possible future manned mission, interest has developed in having a similar type of engineering-oriented atmospheric model as GRAM for the atmosphere of Mars. This report discusses the development of such a new model, the Mars Global Reference Atmospheric Model (Mars-GRAM). The Mars-GRAM program has been developed as a reference model atmosphere for such engineering applications as aerobraking or aerocapture within the atmosphere of Mars (which requires knowledge of Martian atmospheric density to altitudes as low as 20 km), orbiter operations at Mars (requiring atmospheric density information at orbital altitudes), Mars Lander entry and exit operations (requiring atmospheric density and wind estimates from the surface to orbital altitudes), Lander/Rover operational environment on the Mars surface (requiring atmospheric thermal, wind and solar radiation environments at the surface of Mars), and environments for design and planning of balloon measurement systems, such as being considered on the Soviet Mars 1994 Mission (requiring atmospheric thermal, wind and solar radiation environments within the planetary boundary layer of Mars).

This report outlines the observational and modeling basis behind the development of the Mars-GRAM program, provides documentation on the operations of the Mars-GRAM program (available in FORTRAN-77 on IBM-PC compatible 360k diskettes), and presents some example applications for the Mars-GRAM model. The report also suggests some areas for possible improvements for the current Mars-GRAM program, such as the incorporation of routines for the quantitative estimation of solar radiation at the surface of Mars, more realistic treatment of diurnal and latitudinal variations of density, temperature and wind at thermospheric altitudes, and the overall improvement of atmospheric parameterizations by further analysis of the archived data bases for the atmosphere of Mars.

THE MARS GLOBAL REFERENCE ATMOSPHERIC MODEL (MARS-GRAM)

Mars-GRAM, is based on parameterizations to approximate, as realistically as possible, the tmeperature, pressure, density and winds of the Martian atmosphere, and their latitudinal, longitudinal, diurnal, seasonal and altitude variation, from the surface through thermospheric altitudes. Parameterizations are also included for the effects of global-scale dust storms on the variations of the thermodynamic and wind properties of the Martian atmosphere. Recently, David Kaplan, compiler of the definitive report "Environment of Mars, 1988" (Kaplan, 1988), has decided to propose Mars-GRAM as the reference model atmosphere for use by engineers on upcoming equipment design contracts for NASA's Mars Rover and Sample Return Mission (Kaplan, private communication).

The near-surface air temperature on Mars is parameterized in the Mars-GRAM program by computing an approximate value for the geographically and seasonally-dependent daily total absorbed radiation flux. The daily average, maximum and minimum near-surface air temperatures are then calculated from a simple regression relationship assumed between the daily absorbed flux and the temperature parameters. The absorbed flux estimates include the variations in insulation due to the orbital position of Mars, the latitudinal variation of surface albedo (Pollack, et al. 1981), and seasonally-dependent parameterizations for the polar caps (Martin and James, 1986a, 1986b; Iwasaki, 1986; Paige and Ingersoll, 1985; Philip, 1986) and polar hood clouds (James et al., 1987).

The resultant seasonally and latitudinally-dependent daily average, maximum and minimum temperatures from the Mars-GRAM model are shown in Figures 1-3. These agree well with the surface temperature maps of Kieffer et al. (1977), with more realistic variations in the polar regions, as suggested by the observations of Kieffer (1979). Seasonal variations of the daily maximum, minimum and average temperature at the latitudes of the Viking 1 and Viking 2 landers, as evaluated by Mars-GRAM, are shown in Figures 4a and 5a. These plots agree nicely with the Viking 1 and Viking 2 observational data reported by Ryan and Henry (1979), shown for comparison in Figures 4b and 5b.

Parameterizations for the seasonal, latitudinal, diurnal and dust-storm influence on surface pressure were taken from data of Hess et al. (1976, 1977, and 1980), Leovy (1979), Leovy and Zurek (1979), Leovy (1981) and Tillman (1988). Model information on the latitudinal variations of surface pressure was incorporated from Haberle et al. (1982). Figure 6a presents the Mars-GRAM simulations for daily average surface pressure versus time at the Viking 1 and Viking 2 Lander sites. These compare favorably with observations of Hess et al. (1980) and Tillman (1988), as shown in Figure 5b (as presented by Kaplan, 1988). Parameterizations for the seasonal, latitudinal, and diurnal variation of pressure on the reference ellipsoid level were developed from the Viking 1 and Viking 2 data and the model of Haberle et al. (1982). Pressures are adjusted from the values on the reference ellipsoid to values at the local effects of dust storms on daily mean pressure and on the amplitude of the diurnal variation in pressure are evident in Figure 6b. The dust-storm effects on daily average pressure are assumed to be due to perturbations in the meridional circulation strength of the Martian Hadley cell (Haberle et al., 1982), and are therefore taken to have perturbation magnitudes which are antisymmetric in latitude. Thus, when a dust storm causes increased daily average surface pressure at northern latitudes, it is assumed to causes a comparable decrease in daily average surface pressure at the corresponding southern latitude. For lack of any other information, the effects of dust storms on the magnitude of the diurnal variation are assumed to be symmetric in latitude (i.e. equal in magnitude and sign at comparable northern and southern latitudes).

Mars-GRAM parameterizations for the geographical, seasonal and altitude dependence on non-dust storm, daily average temperatures above the surface come from a combination of observational and model values. The observational data include: Mariner 9 IRIS data from Conrath (1981), and Conrath data as reported by Leovy (1982) and Magalhaes (1987), Viking IRTM data (Martin et al., 1982), Mariner 9 radio occultation data from Kliore et al. (1983), and Viking 1 radio occultation data (Fjeldbo et al., 1977), Lindal et al. (1979), and Davies (1979). Model output used include results from Pollack et al. (1981), Haberle et al. (1982), and Pitts et al. (1988).

The parameterizations in Mars-GRAM for temperature variations during dust storm conditions were taken from observational data of Kliore et al. (1972), Jakosky and Martin (1987), Martin et al. (1982), Hanel et al. (1972), and Conrath (1975). Model values for dust storm effects were taken from Haberle et al. (1982) and Pitts et al. (1988).

Data for the Mars-GRAM parameterizations of the height variation of the diurnal (longitudinal) variations of temperature about the daily mean value were taken from Viking IRTM observations of Martin et al. (1982), Mariner 9 IR spectroscopy data of Hanel et al. (1972), and model results from Zurek, as reported in Pitts et al. (1988). Diurnal temperature variations are significantly larger during dust storm periods than during non-dust storm conditions.

Examples of the altitude dependence of temperature and density during non-dust storm conditions are provided by Figures 7 and 8, which compare Mars-GRAM simulations for the date, time and location of the Viking 1 Lander site with temperature observations from the Viking Lander 1 entry profile (Seiff and Kirk, 1977) and the COSPAR Northern Hemisphere Summer Mean density profile (Seiff, 1982). Mars-GRAM simulations of the height and latitudinal variation of daily average temperatures are shown in Figure 9, for Northern Hemisphere spring equinox (areocentric longitude of sun, $L_s = 0°$), in Figure 10 for Northern Hemisphere summer solstice ($L_s = 90°$), and in Figure 11 for Northern Hemisphere winter solstice ($L_s = 270°$). Examples of Mars-GRAM simulations of dust storm effects on surface temperature are shown in Figures 4a and 5a. An example of a Mars-GRAM dust-storm height-latitude temperature cross section (at $L_s = 270°$) is shown in Figure 12.

In Figures 13 and 14 the model height-latitude cross sections of temperature and zonal wind are compared for Mars-GRAM simulation results for $L_s = 49°$ and for Mariner 9 results from L_s = 43-54° (Leovy, 1982, from data provided by Conrath). For the same conditions as in Figure 13, Figure 15 shows Mars-GRAM simulations of height-latitude cross sections of pressure (N/m^2, log-base-10 scale) and of density (kg/m^3, log-base-10 scale). Figure 15a illustrates the fact that the conversion between height and pressure is not constant with latitude at a given seasonal time.

The perturbations in pressure and density evident between about 30°S and 60°S in Figure 15 are due to the influence of terrain height on surface pressure. The cross sections of Figure 15 were evaluated along the zero longitude meridian. Terrain height contours used in Mars-GRAM, shown in Figure 16, indicate that surface altitudes reach more than 3 km above the reference ellipsoid over the latitude range 30-60°S. The reference ellipsoid in Mars-GRAM is defined by the equatorial and polar radii as given by Cain et al. (1973).

Examples of the Mars-GRAM capabilities to simulate the spatial and temporal variation of global dust storm effects on temperature are provided by Figures 17 and 18. These figures show both daily average temperatures and the range of diurnal variation in temperature during the development (Figure 17) and decay (Figure 18) of global dust storms. Figure 17, similar in format to Figure 5 of Martin et al. (1982), shows the simulated development of dust storm effects on temperature at 30 km altitude for a simulation of the 1977b global dust storm, which was observed by Viking IRTM. The simulations in Figure 17 were based on an assumed intensity of 3.0 (the maximum possible in Mars-GRAM). Comparison with the data of Martin et al. indicates that better correspondence would be achieved by attributing an intensity of about 2.0 to the 1977b storm. Figure 18, similar in format to Figure 1 of Conrath (1975), is for a simulation of the decay of the 1971b global dust storm, which was observed by Mariner 9. The simulation in Figure 18 also assumed an intensity of 3, although the model would compare better with the 1971b storm observations if an intensity of about 2 were assumed.

The newly-developed parameterizations for Mars-GRAM provide a simulation capability to altitudes reaching the base of the thermosphere. For simulations of the seasonal, geographical and solar-activity dependence of thermospheric conditions, the Mars-GRAM uses an adaptation of the thermospheric model of Stewart (Culp and Stewart, 1983, 1984; Stewart and Hanson, 1982; Stewart, 1987; see also the revised Stewart program code given in Pitts et al., 1977).

The Stewart model thermosphere incorporates results from a number of data and model sources, e.g. the Mars Reference Atmosphere (Kliore, 1982), occultation data and mass spectrometer data from Mariner and from Viking orbiters (Fjeldbo et al., 1966, 1970, 1977; Kliore et al., 1972; Stewart et al., 1972; Nier and McElroy, 1977), and data from the Viking lander atmospheric entry trajectories (Seiff and Kirk, 1977). It includes parameterizations to simulate the effects of solar activity, seasonal variation, diurnal variation magnitude, dust storm effects, and effects due to the orbital position of Mars.

MARS-GRAM PROFILES OF MEAN TEMPERATURE, PRESSURE AND DENSITY

Parameterization for the representation of surface temperature and vertical temperature profiles, based on summaries from observational and model data, are discussed in the previous section. The addition of a parameterization for the variations in surface pressure, also discussed in the previous section, then allows for computation of vertical profiles of pressure and density by use of the hydrostatic and perfect gas law relations.

Temperature profiles are built up from parameterizations of the surface temperature and temperature lapse rates between the significant levels set at 5, 15, 30, 50 and 75 km. Parameterization of the spatial and temporal variations of surface temperature were discussed in the previous section. From the summaries of the observational and model data, the spatial and temporal variations of the lapse rates between the significant levels were developed. Temperature profiles are further constrained to be consistent with seasonal and latitudinal variations near 25 km altitude, as presented in Figure 14 of Leovy (1982).

The finally-adjusted temperature lapse rates, appropriate to the specified time and position, are assumed to be constant between each of the significant height levels. With the specification of surface pressure to serve as a boundary condition, the pressure profile is easily computed by integration of the hydrostatic relationship, $dp/dz = -\rho g = -pg/RT$, over each of the vertical sections of the constant lapse rate. After a solution for the pressure is obtained, the density is determined from the temperature and pressure by the perfect gas law, $\rho = p/RT$. Here R is the gas constant for a gas of the composition of the atmosphere of Mars and g is the altitude-dependent value of the acceleration of gravity on Mars (Seiff, 1982).

The Stewart model thermosphere starts with a specification of the temperature T_F at the base of the thermosphere (at a height Z_F). Both T_F and Z_F depend on the orbital position of Mars. Between 75 km altitude and height Z_F, the temperature profile is constructed from a combination of constant lapse rate and constant temperature segments, as necessary to preserve the hydrostatic relationship for pressure between 75 km and height Z_F (where the Stewart model specifies the pressure to be 1.26 X 10^{-4} N/m^2). If T_F is greater than T(75 km), then the temperature profile between 75 km and Z_F has a constant temperature [= T(75km)] in the lower part and a constant lapse rate in the upper rate. If T_F is less than T(75 km), then the temperature profile between 75 km and Z_F is taken to have a constant lapse rate in the lower part and a constant temperature (= T_F) in the upper part.

TERRAIN EFFECTS ON TEMPERATURE PROFILES

A problem in the Martian atmosphere is how to represent the effects of terrain variations on temperature profiles. This problem, as stated by Seiff (1982) is:

> "Near the surface, it is necessary to refer the profile to altitude above terrain, as the terrain controls the near-surface temperatures. At higher altitudes also, it is indicated that radiative equilibrium is the first order control over temperature (Seiff and Kirk, 1977) with the input at the lower boundary being radiation from the surface. The surface temperature, in turn, is controlled by solar flux and not the surface elevation. Hence, we have assumed in the model that over extensive regions of depressed or elevated terrain ... the profile [of temperature versus height up to 100 km] ... should be applied at altitudes measured above the terrain. For more local depressions or elevations, the model should be entered at an altitude measured relative to the mean surrounding surface."

Simply put, the Martian air on the mountain tops will not be nearly as cold compared to the air in the valleys as is the case on Earth. Not only is this due to the increased importance of surface radiation, but also due to the fact that the air density is low and airflow over the mountains will not be as effective in moderating the effect of surface radiation on the mountaintop surface to a temperature near that of the approaching, high altitude, cool air flow.

Mars-GRAM treats this problem, as does Seiff, by assuming that the surface temperature is unaffected by terrain variations, and that the vertical temperature profile, relative to the surface temperature, must be specified as a function of height above the local terrain surface.

An example of the interrelationship between surface temperature and temperature profile is seen in the present Mars-GRAM results near the south pole in Figure 11. The strong latitudinal gradient in surface temperature near the polar cap edge is consistent with observations and other models. The corresponding latitudinal gradient in upper level temperatures (e.g. the 25 km temperature of Figure 14 of Leovy, on which Mars-GRAM is based) is not observed to be so large. Therefore there must be a strong latitudinal gradient in near-surface temperature lapse rates in this region near the edge of the polar cap. Details of the effects of surface radiation and terrain elevation on the near-surface temperature profile is an atmospheric boundary layer problem. The boundary layer of the Martian atmosphere has been studied only relatively little (see Sutton et al., 1978, and Kieffer et al., 1976). This problem of temperature effects on near surface temperatures and lapse rates is one which requires further examination in the future.

WIND PROFILES IN MARS-GRAM

Wind components are evaluated in Mars-GRAM by the geostrophic (or, on Mars, areostrophic) approximation, obtained by a steady balance between pressure gradient and Coriolis forces, namely:

$$u_a = -(1/\rho f)\partial p/\partial y \qquad \text{and} \qquad v_a = (1/\rho f)\partial p/\partial x, \tag{1}$$

where f is the Coriolis parameter [$2\Omega \sin(\theta)$, where Ω is the planetary rotation rate and θ is the latitude], ρ is density, and the pressure gradient components are evaluated by finite differences from pressures evaluated at ±2.5 degrees of latitude and longitude from the position at which the wind components are evaluated. Since the areostrophic wind relations (1) follow the thermal wind balance, wind evaluated by these equations, from local horizontal pressure radients, are consistent with winds evaluated by vertical integration of the thermal wind equations (e.g. compare Figures 13b and 14).

Following Fleming et al. (1988), we take the winds near the equator to be given by

$$u_a = -\{a/[2\,\rho\,\Omega\cos(\theta)]\}\partial^2 p/\partial y^2 \ , \tag{2}$$

and
$$v_a = \{a/[2\,\rho\,\Omega\cos(\theta)]\}\partial^2 p/\partial x\partial y\ , \tag{3}$$

where a is the radial distance from the planetary center, and the winds near the poles ($|\theta| > 75°$) are computed from

$$u_a \cos(\theta)/\cos(75°) \qquad \text{and} \qquad v_a \cos(\theta)/\cos(75°)\ . \tag{4}$$

Smooth transition between (1), (2) and (3) is accomplished by interpolation between $\theta = 7.5°$ [at which (2) and (3) apply] and $\theta = 15°$ [at which (1) applies].

At high altitudes, where the density ρ becomes low, the areostrophic winds become large because ρ is in the denominator of equations (1)-(4). To overcome this problem molecular viscosity terms are added to the momentum balance equations, of the form

$$-\nu\,\partial^2 u/\partial z^2 \cong -\nu\, u/L^2 \qquad \text{and} \qquad -\nu\,\partial^2 v/\partial z^2 \cong -\nu\, v/L^2\ , \tag{5}$$

where $\nu = \mu/\rho$, μ is the molecular viscosity, and L is a viscous gradient scale (parameterized currently as being equal to the pressure scale height H). With the addition of the viscous terms in (5) to the balance between pressure gradient and Coriolis force terms, the viscous-modified areostrophic wind components become

$$u_v = [f^2 u_a - (\nu/L^2) f v_a] / [f^2 + (\nu/L^2)^2], \tag{6a}$$

$$v_v = [f^2 v_a + (\nu/L^2) f u_a] / [f^2 + (\nu/L^2)^2]\ . \tag{6b}$$

At sufficiently high altitudes that $(\nu/L^2) >> f$, the viscous-modified winds become

$$u_v = -(L^2/\mu)\partial p/\partial x \qquad \text{and} \qquad v_v = -(L^2/\mu)\partial p/\partial y\,. \tag{7}$$

The winds evaluated by (7) do not suffer the problem of having density in the denominator, and so do not become too large. The winds evaluated by (7) are orthogonal to the ordinary areostrophic wind components, that is they are parallel to the pressure gradients instead of parallel to the isobars. This approximation is borne out qualitatively by Figure 2 of Bougher et al. (1988), which shows that, on a constant pressure surface, the winds are approximately perpendicular to the isotherms, and hence would be approximately parallel to the pressure gradients on a constant height surface.

Unfortunately the parameterization of equation (6) cannot be tested thoroughly at the present, due to the limitations of the Stewart thermosphere model. Since Stewart's model is intended to be primarily a global mean thermosphere, the horizontal gradients of pressure (as well as those of density and temperature) are too small to be realistic. Thus winds evaluated in the present Mars-GRAM are significantly smaller in magnitude than those produced by the realistic Martian thermospheric model of Bougher et al. (1988).

Near the terrain surface, a simplified boundary layer model for winds is included in Mars-GRAM. In this model the boundary layer winds are estimated by

$$u(z) = [(C_D)^{1/2} / k] \log(z/z_o)\, u_a \quad , \quad \text{and} \tag{8}$$

$$v(z) = [(C_D)^{1/2} / k] \log(z/z_o)\, v_a \quad , \tag{9}$$

where the drag coefficient is estimated to be 0.0015, the von Karman constant = 0.4, and the surface roughness is estimated to be 0.03m. If $z < 1.6$ m (the height of the Viking Lander "surface" wind measurements), then z is taken to be 1.6 m in equations (8) and (9). The height at which equations (8) and (9) produce winds equal to u_a and v_a is taken to be the top of the boundary layer.

WAVE PERTURBATION MAGNITUDES IN MARS-GRAM

Wave-like perturbations have been observed in the Viking 1 and Viking 2 surface pressure data (Leovy, 1981), in the Mariner 9 IR spectroscopy data (Pirraglia and Conrath, 1974; Conrath, 1976, 1981), and the Viking 1 and Viking 2 lander entry profiles (Seiff and Kirk, 1976, 1977). Most of the wave-like perturbations have been interpreted as atmospheric tides (Pirraglia and Conrath, 1974; Conrath, 1976; Seiff, 1982), or as planetary waves (Conrath, 1981; Barnes and Hollingsworth, 1988). The theory of thermal tides in the Martian atmosphere has been well explored (Zurek, 1976, 1986; Zurek and Haberle, 1988). Compared to the detailed tidal analyses, relatively little attention has been paid to other forms of wave perturbations such as gravity waves (Gadian and Green, 1983), Kelvin waves and normal modes (Zurek, 1988; 1979). The very strong influence of topography on the wave structure in the Martian atmosphere has been documented both by observational data (Conrath, 1976) and by theoretical analysis (Zurek, 1976).

One of the best-known properties of gravity waves is that their amplitudes increase with height as the inverse square root of the mean atmospheric density. In this way, the kinetic energy of the wave is maintained even as the packet propagates upward into the rarer regions. The effect has been likened to the wave-kink that grows as it travels along a whip from the thick end near the handle to the thin region near the tip. The comparison is even more apt than first appears, for just as the reach heights at which it becomes too large for smooth fluid behavior and it turns over into shocks and/or turbulence.

One of the common sources of gravity waves is wind flow over mountains. As the atmospheric flow reaches the mountain, surface pressures deflect it up and over the mountain rise, and then restore it to level on the other side. However, the disturbance transmitted to the air flow is not confined to some surface region, as the pressure fields associated with the deflection have space and time scales that match the characteristics of gravity waves in the free atmosphere. Distorting any elastic medium in a way that matches its natural modes causes the excitation of waves that broadcast energy from the disturbance to the far reaches of the medium. So, around mountainous regions the atmosphere normally contains a wealth of wave activity. Mountain waves, especially the resonant forms known as lee waves are a familiar phenomenon of the Earth's atmosphere, and are expected to be of even more importance in the Martian atmosphere.

Because erosion is much weaker in the rainless Martian environment, and because surface gravity is lower, surface features on Mars tend to be more severe than on Earth. Ridges and mountains are higher, while surface winds are comparable (and, at times, higher), so the Martian mountain waves provide stronger aerodynamic disturbances than are encountered on Earth.

The waves of most interest for aerobraking or lander vehicle entry are readily modeled by the long-wave approximation. This component of the disturbance maps up through the atmosphere as a replica of the underlying terrain, and, in common with all gravity waves, the associated fractional density disturbances grow ever greater with altitude. Effectively the probing spacecraft will be "flying into the mountains", even when it is several tens of kilometers above the peaks. Limits may be imposed by wave saturation (Smith et al., 1987), absorption by critical layers (Hines, 1960), or by dissipation of the waves by molecular viscosity (Pitteway and Hines, 1963), but over the larger terrain features the tops of these ghost mountains may occur as shocks or wedges of turbulence.

In the long-wave approximation, we assume that the wave number k satisfies the inequality

$$k^2 \ll N^2/U^2 \quad , \tag{10}$$

where N is the Brunt-Vaisala frequency, and U is the mean wind speed. Under this assumption the governing differential equation for wave perturbations is

$$d^2y/dz^2 + [N^2/U^2 - (d^2U/dz^2)/U - 1/4H^2]y(z) = 0 \quad , \tag{11}$$

where H is the atmospheric scale height. The Fourier transform, $\mathbf{w}(z)$, for the vertical velocity perturbations is, for example, given by

$$\mathbf{w}(z) = \rho_o(z)^{-1/2} A(k)\, y(z) \quad , \tag{12}$$

where ρ_o is the mean density, A(k) is presently an unknown Fourier amplitude, and y(z) is the solution obtained from equation (11).

The assumption of purely tangential flow for the surface boundary condition determines the wave amplitude in terms of the slope dh/dx of the surface terrain and the mean wind speed U(0) that must be deflected over the terrain undulation. The Fourier transform of this surface condition defines the component amplitudes A(k) of equation (12) through the Fourier components of the surface terrain shape, h(x).

From this approach, the Fourier transform of the density variations can be obtained. Since there is no k dependence for the coefficients relating the Fourier transform of the density to the Fourier transform of the terrain surface, the Fourier inversion to obtain the vertical profile of density perturbations is straightforward, and yields

$$\rho_w(x,z)/\rho_o = \left[\frac{y(z)U(0)N^2}{y(0)U(z)g}\left(\frac{\rho_o(0)}{\rho_o(z)}\right)^{1/2}\right]h(x) \quad , \qquad (13)$$

The implications of equation (13) are clear:

(a) The horizontal structure of the density disturbance ρ_w is an image of the terrain shape h(x).

(b) There is the standard vertical growth factor associated with gravity waves, proportional to the inverse square root of the mean density.

(c) There is a vertical wave structure y(z)/y(0), which will usually be closely sinusoidal.

(d) There is an excitation factor $[U(0)N^2/U(z)g]$ relating surface conditions to the generated wave density field.

These results apply equally to an isolated surface feature, such as a mountain sitting alone in the plains region, or an extended complex area of severe terrain. If the surface shape is known, and one differential equation involving the meteorological functions U(z) and N(z) is solved, the density perturbation, $\rho_w(x,z)$, at any location above the surface is obtained. Although derived here for the two-dimensional case, the results of equation (13) can easily be generalized to a fully three-dimensional terrain description.

If non-linear processes do not limit the growth of the wave, the high kinematic molecular viscosity of the upper atmosphere will. Because of lower density, the molecular viscosity effect will be felt at lower altitudes on Mars than on Earth. The treatment of Pitteway and Hines (1963), can be used to model this effect.

If a WKB approximation solution is applied for (11) and (13), the amplitude of the density perturbations due to the mountain waves can be approximated as

$$|\rho_w(x,z)|/\rho_o = \left[\frac{N(0)N(z)}{g}\left(\frac{\rho_o(0)}{\rho_o(z)}\right)^{1/2}\right]h(x) \quad . \qquad (14)$$

This form of the solution eliminates the explicit dependence on U(z) and U(0), so that all of the terms in (14) are readily evaluated by Mars-GRAM from the terrain height data and the thermodynamic parameterization models. This relation [with the terrain relief h(x) somewhat exaggerated because of the present coarse resolution used] is employed in Mars-GRAM to estimate the magnitude of the random density perturbation component below 75 km altitude.

In order to insure that the magnitudes estimated by (14) do not become too large [because of the $\rho_o(z)$ term in the denominator], a viscous damping model must also be applied here, as well as to the areostrophic wind components. Following the approach of Pitteway and Hines (1963), an approximate solution for the damping factor (appropriate in the long-wave limit), to be applied as a factor to equation (14), is obtained as

$$\exp\{-[L_h \, \nu \, N^3 \, H/(4\pi \, U^4) \,] \, [1 - \rho_o(z) \, / \, \rho_o(0)]\} \quad , \tag{15}$$

where L_h is the horizontal wavelength of the dominant mode of the mountain waves. L_h is taken to be the same as the horizontal scale used in the density perturbation model (see next section).

Relations (14) and (15) are used to estimate the density perturbation magnitudes up to a height of 75 km. Above the base of the thermosphere (at height Z_F), the short-term density perturbation magnitudes of the Stewart model are employed. Between 75 km and Z_F, interpolation is used to insure a smooth transition between these two height regions.

DENSITY PERTURBATION SIMULATION IN MARS-GRAM

The same density perturbation model used in the Earth GRAM program can be utilized in Mars-GRAM, provided that the perturbation magnitudes, horizontal scales, and vertical scales are adapted to values applicable for the Martian atmosphere. The density perturbation model estimates relative density perturbations $\rho'(x,y,x,t)$ by the relation

$$\rho'(x+\Delta x,y+\Delta y,z+\Delta z,t+\Delta t) \;=\; \alpha\,\rho'(x,y,x,t) + \beta\,R(x,y,z,t) \quad , \tag{16}$$

where α is the correlation $r(\Delta x, \Delta y, \Delta z)$ over the spatial separation involved between steps, given by

$$r(\Delta x, \Delta y, \Delta z) \;=\; \exp\{-[(\Delta x/L_h)^2 \;+\; (\Delta y/L_h)^2 \;+\; (\Delta z/L_z)^2]^{1/2}\} \quad , \tag{17}$$

β is given by $[1 - \alpha^2]^{1/2}$, and $R(x,y,z,t)$ is a normally-distributed uncorrelated random sequence, having the same rms magnitude as the relative density perturbations ρ'.

Evaluation of the magnitudes of the density perturbations used in Mars-GRAM were discussed in the previous section. The vertical scale for the density perturbations, L_z, was estimated from structure-function analysis of the Viking 1 and Viking 2 density profiles (Seiff and Kirk, 1977) to be 8 km, at least for altitudes up to 100 km. For the horizontal scale of the density perturbations, L_h, the values used in the Earth GRAM model were scaled by a factor of 1.5, as estimated to be applicable for the Martian atmosphere. Namely, in Mars-GRAM we take

$$L_h \;=\; 30 \;+\; 0.01875\, z^2 \quad , \tag{18}$$

with z and L_h in km. An upper limit of 600 km is imposed for L_h at high altitudes.

CONCLUSIONS

The Mars-GRAM has numerous applications as a "poor man's global circulation model". For example, the computation of all of the data necessary to describe the complete seasonal variations at the surface (Figures 1-3) and all altitudes (e.g., Figure 9), takes at most a few minutes on an IBM-PC (with 8087 co-processor; even faster on an 80286-based machine, such as an IBM-AT or compatible, or on an 80386-based machine). Comparable data would take many hours of computation on a mainframe using a 3-D global circulation model for Mars. The diurnal (longitudinal) variability incorporated into the Mars-GRAM program is not even available in 2-D version of a Mars global circulation model.

In addition to the engineering applications envisioned for Mars-GRAM (e.g., aerocapture mission profile studies, Mars Rover Sample Return mission planning and design), the Mars-GRAM has a number of potential scientific applications. One of these is its ability to provide a realistic, geographically and seasonally-dependent background of temperatures and winds for studies of tides and the atmospheric propagation of other wave disturbances (e.g., gravity waves, mountain lee waves, etc.). Another application would be in providing realistic "first guess" profiles for the inversion processing for temperature retrievals from temperature sounders on upcoming Mars missions.

Of course, being a parameterization model, rather than a first principles one such as a global circulation model, Mars-GRAM is only as good as the parameterizations built into it. Also it cannot test the sensitivity to variation of parameters beyond those on which the parameterizations are based (e.g., it cannot estimate the effects of a dust storm of twice the optical depth previously observed). With continued analysis of additional observational data from the Viking and Mariner programs, analysis of new results from global circulation models, and with new data expected to be coming in from the Mars Observer program, Mars-GRAM should steadily improve in its realism and reliability in the future.

PLANS FOR FUTURE IMPROVEMENTS IN MARS-GRAM

Although the goal of producing a fairly realistic model of the Martian atmosphere has been realized in the current version of Mars-GRAM, there is still room for substantial improvement and enhancement of the program. There is a considerable volume of computerized data, from the Mariner and Viking programs, available at government and university planetary studies laboratories around the country. As discussed in the previous section, since the current Mars-GRAM parameterizations were developed only from the limited amount of data and information in the open literature, there is room for considerable improvement in these parameterizations by analysis of these considerable-sized computer data bases.

As discussed in the section of winds in the thermosphere, there is considerable room for improvement in the horizontal gradients of density, temperature and pressure in the thermospheric model. Simple parameterizations of the latitudinal and longitudinal (diurnal) variations of temperature could be developed from the realistic thermospheric simulations of the model of Bougher et al. (1988) (e.g., see their Figures 3 and 4). The current approach of the Stewart thermospheric model should then be adequate for estimating density and pressure from these improved thermospheric temperature estimates.

Since the current Mars-GRAM relies on estimates of the daily total surface solar irradiance for the estimation of daily maximum, average and minimum temperatures, it would be a straightforward addition to Mars-GRAM to add a subroutine which calculates the direct (beam) and diffuse (scattered) components of the time-dependent solar irradiance at the surface. Such a routine for estimation on the surface solar irradiance would be based on the model approach of Justus and Paris (1985), with parameters adapted for conditions of the Mars atmosphere and dust optical properties. The influence on solar irradiance due to dust storm conditions could easily be incorporated, since optical depth has been measured at the Viking Lander sites (Tillman et al., 1979; Zurek, 1982), and other dust optical properties have also been inferred (Toon et al., 1977). Addition of a surface solar irradiance module for Mars-GRAM would be very useful for such applications as: (1) analysis of thermal heating environments and solar power cell performance on surface-based systems, such as Mars Rover, (2) analysis of heating, life and trajectory for balloon systems, such as that proposed for the Soviet Mars 1994 mission, and (3) for general improvements in the capability to parameterize and study the Planetary Boundary Layer of Mars and its energy and momentum flux budgets.

Users or potential users of Mars-GRAM who are interested in the possibility of one or more of these modifications being incorporated into the program are asked to contact either Dr. Justus at Computer Sciences Corporation or Mr. Dale Johnson at NASA Marshall Space Flight Center.

ACKNOWLEDGMENTS

The authors wish to express appreciation to Mr. Joe Gamble and Dr. David Kaplan, of NASA Johnson Space Center for their encouragement (and monetary support) of the Mars-GRAM development program. Thanks are also due to Mr. Omar Torres, of Georgia Tech, who provided valuable assistance in reading data from graphs in the published literature and in the development of the surface pressure parameterizations used in Mars-GRAM.

REFERENCES

Barnes, J. R., and J. L. Hollingsworth (1988). Dynamical Modeling of a Planetary Wave Mechanism for a Martian Polar Warming. Icarus, 71(2): 313-334

Bougher, S. W., et al. (1988). Mars Thermospheric General Circulation Model: Calculations for the Arrival of Phobos at Mars. Geophys. Res. Let., 15(13): 1511-1514

Cain, D. L., A. J. Kliore, B. L. Seidel, M. J. Sykes, and P. Woiceshyn (1973). Approximations t the Mean Surface of Mars and Mars Atmosphere Using Mariner 9 Occultations. J. Geophys. Res., 78(20): 4352-4354

Conrath, B. J. (1975). Thermal Structure of the Martian Atmosphere During the Dissipation of the Dust Storm of 1971. Icarus, 24: 36-46

Conrath, B. J. (1976). Influence of Planetary-Scale Topography on the Diurnal Thermal Tide During the 1971 Martian Dust Storm. J. Atmos. Sci., 33: 2430-2439

Conrath, B. J. (1981). Planetary-Scale Wave Structure in the Martian Atmosphere. Icarus, 48: 246-255

Culp, R. D., A. I. Stewart, C. Chow (1983). Time Dependent Model of the Martian Atmosphere for Use in Orbit Lifetime and Sustenance Studies. JPL Contract #956446: Univ. of Colorado, 31 pp.

Culp, R. D., and A. I. Stewart (1984). Time-Dependent Model of the Martian Atmosphere for Use in Orbit Lifetime and Sustenance Studies. J. Astron. Sci., 32(3): 329-341

Davies, D. W. (1979). The Relative Humidity of Mars' Atmosphere. J. Geophys. Res., 84(B14): 8335-8340

Fjeldbo, G. et al. (1966). Atmosphere of Mars: Mariner IV Models Compared. Science, 153: 1518-1523

Fjeldbo, G. et al. (1970). The Mariner 1969 Occultation Measurements of the Upper Atmosphere of Mars. Radio Science, 5: 381-386

Fjeldbo, G., D. Sweetnam, J. Brenkle, E. Christensen, D. Farless, et al (1977). Viking Radio Occultation Measurements of the Martian Atmosphere and Topography: Primary Mission Coverage. J. Geophys. Res., 82(28): 4317-4324

Fleming, E. L et al. (1988). Monthly Mean Global Climatology of Temperature Wind, Geopotential Height, and Pressure for 0-120 km. NASA Technical Memorandum 100697

Gadian, A. M., and J. S. A. Green (1983). A Theoretical Study of Small Amplitude Waves in the Martian Lower Atmosphere and a Comparison Made with Those on Earth. Annales Geophysicae, 1(3): 239-244

Haberle, R. M., C. B. Leovy, and J. B. Pollack (1982). Some Effects of Global Dust Storms on the Atmospheric Circulation of Mars. Icarus, 50: 322-367

Hamilton, K. and R. R. Garcia (1986). Theory and Observations of the Short-Period Normal Mode Oscillations of the Atmosphere. J. Geophys. Res., 91(D11): 11,867-11,875

Hanel, R., B. Conrath, W. Hovis, V. Kunde, P. Lowman, W. Maguire, J. Pearl, et al. (1972). Investigation of the Martian Environment by Infrared Spectroscopy on Mariner 9. Icarus, 17: 423-442

Hess, S. L. et al. (1976) Mars Climatology from Viking 1 After 20 Sols. Science, 194: 78-80

Hess, S. L. et al. (1977). Meteorological Results of the Surface of Mars: Viking 1 and 2. J. Geophys. Res., 82: 4459-4574

Hess, S. L. et al. (1980). The Annual Cycle of Pressure on Mars Measured by Viking Landers 1 and 2. Geophys. Res. Lett., 7: 197-200

Hines, C. O. (1960). Internal Atmospheric Gravity Waves at Ionospheric Heights. Canadian J. Phys., 38: 1441

Iwasaki, K. et al. (1986). Interannual Differences of Mars Polar Caps. MECA Symposium on Mars: Evolution of Its Climate and Atmosphere. LPI Tech. Rept., 87.01: 58-59

Jakosky, B. M. and T. Z. Martin (1987). Mars: North-Polar Atmospheric Warming During Dust Storms. Icarus, 72: 528-534

James, P. B., M. Pierce, L. J. Martin (1987). Martian North Polar Cap and Circumpolar Clouds: 1975-1980 Telescopic Observations. Icarus, 71: 306-312

Justus, C. G. and M. V. Paris (1985). A Model for the Solar Spectral Irradiance and Radiance at the Bottom and Top of a Cloudless Atmosphere. J. Climate Appl. Meteorol., 24(3): 193-205

Justus, C. G., R. G. Roper, Arthur Woodrum, and O. E. Smith (1975). Global Reference Atmospheric Model for Aerospace Applications. J. Spacecraft and Rockets, 12: 449-450

Justus, C. G., R. G. Roper, Arthur Woodrum, and O. E. Smith (1976). A Global Reference Atmospheric Model for Surface to Orbital Altitudes. J. Appl. Meteorol., 15: 3-9

Justus, C. G. and R. G. Roper (1987). Application of the Global Reference Atmospheric Model to Polar Orbit Missions. AIAA 25th Aerospace Sciences Meeting, January, Reno, NV, paper AIAA-87-0264

Justus, C. G. (1988). Density Perturbation Simulation with the Global Reference Atmospheric Model. AIAA 26th Aerospace Science Meeting, January, Reno, NV, paper AIAA-88-0494

Kaplan, David (1988). Environment of Mars, 1988. NASA Tech. Memo 100470, October, 62 pp.

Kieffer, H. H. (1979). Mars South Polar Spring and Summer Temperatures: A Residual CO_2 Frost. J. Geophys. Res., 84 (B14): 8263-8288

Kieffer, H. H., P. R. Christiansen, T. Z. Martin, E. D. Miner, and F. D. Palluconi (1976). Science, 194: 1346-1351

Kieffer, H. H., T. Z. Martin, A. R. Peterfreund, B. M. Jakosky, E. D. Miner, and F. D. Palluconi (1977). Thermal and Albedo Mapping of Mars During the Viking Primary Mission. J. Geophys. Res., 82: 4249-4291

Kliore, A. (editor) (1982). The Mars Reference Atmosphere. Adv. Space Res., 2

Kliore, A. J., D. L. Cain, G. Fjeldbo, B. L. Seidel, M. J. Sykes, S. I. Rasool (1972). The Atmosphere of Mars from Mariner 9 Radio Occultation Measurements. Icarus, 17: 484-516

Kliore, A. J., G. Fjeldbo, B. L. Seidel, M. J. Sykes, and P. M. Woiceshyn (1973). S Band Radio Occultation Measurements of the Atmosphere and Topography of Mars with Mariner 9: Extended Mission Coverage of Polar ... J. Geophys. Res., 73(20): 4331-4351

Leovy, C. B. (1979). Martian Meteorology. Ann. Rev. Astron. Astrophys., 17: 387-413

Leovy, C. B. (1981). Observations of Martian Tides Over Two Annual Cycles. J. Atmos. Sci., 38: 30-39

Leovy, C. (1982). Martian Meteorological Variability. Adv. Space. Res., 2: 19-44

Leovy, C. B. and R. W. Zurek (1979). Thermal Tides and Martian Dust Storms: Direct Evidence for Coupling. J. Geophys. Res., 84: 2956-2968

Lindal, G. F., et al. (1979). Viking Radio Occultation Measurements of the Atmosphere and Topography of Mars: Data Acquired During 1 Martian Year of Tracking. J. Geophys. Res., 84: 8443

Magalhaes, J. A. (1987). The Martian Hadley Circulation: Comparison of "Viscous" Model Predictions to Observations. Icarus, 70: 442-468

Martin, L. J. and P. B. James (1986a). Major Dust Storm Activity and Variations in the Recession of Mars' South Polar Cap. MECA Workshop on the Evolution of the Martian Atmosphere, LPI Tech. Rept. 86-07: 29-30

Martin, L. J. and P. B. James (1986b). The Great Dust Storm of 1986(?). MECA Workshop on Mars: Evolution of Its Climate and Atmosphere. LPI Tech Rept. 87-01: 76-77

Martin, T. Z., M. M. Kieffer, and E. D. Miner (1982). Mars' Atmospheric Behavior from Viking Infra-red Thermal Mapper Measurements. Adv. Space Res., 2: 57-65

Nier, A. O. and M. B. McElroy (1977). Composition and Structure of Mars' Upper Atmosphere: Results from neutral Mass Spectrometer on Viking 1 and 2. J. Geophys. Res., 82: 4341-4350

Paige, D. A. and A. P. Ingersoll (1985). Annual Heat Balance of Martian Polar Caps: Viking Observations. Science, 228: 1160-1168

Philip, J. R. (1986). Similarity Analysis of the Martian Polar Caps. Geophys. Res. Lett., 13(11): 1137-1140

Pickersgill, A. O. and G. E. Hunt (1979). The Formation of Martian Lee Waves Generated by a Crater. J. Geophys. Res., 84: 8317-8331

Pirraglia, J. A. and B. J. Conrath (1974). Martian Tidal Pressure and Wind Fields Obtained from the Mariner 9 Infrared Spectroscopy Experiment. J. Atmos. Sci., 31: 318-329

Pitteway, M. L. V. and C. O. Hines (1963). The Viscous Damping of Atmospheric Gravity Waves. Can. J. Phys., 41: 1935

Pitts, D. E., J. E. Tillman, J. Pollack, R. Zurek (1988). Model Profiles of the Mars Atmosphere for the Mars Rover and Sample Return Mission. Preprint

Pollack, J. B., C. B. Leovy, P. W. Greiman, and Y. Mintz (1981). A Martian General Circulation Experiment with Large Topography. J. Atmos. Sci., 38(1): 3-29

Ryan, J. A. and R. M. Henry (1979). J. Geophys. Res., 84: 2821-2829

Seiff, A. (1982). Post-Viking Models for the Structure of the Summer Atmosphere of Mars. Adv. Space. Res., 2: 3-17

Seiff, A. and D. B. Kirk (1976). Structure of Mars' Atmosphere up to 100 Kilometers from the Entry Measurements of Viking 2. Science, 194: 1300-1302

Seiff, A. and D. B. Kirk (1977). Structure of the Atmosphere of Mars in Summer at Mid-Latitudes. J. Geophys. Res., 82(28): 4364-4378

Smith, S. A., D. C. Fritts, and T. E. Van Zandt (1987). Evidence of a Saturation Spectrum of Gravity News. J. Atmos. Sci., 44(12): 1404-1410

Stewart, A., et. al. (1972). Mariner 9 Ultraviolet Spectrometer Experiment: Structure of Mars' Upper Atmosphere. Icarus, 17: 469-474

Stewart, A. I. F. (1987). Revised Time Dependent Model of the Martian Atmosphere for use in Orbit Lifetime and Sustenance Studies. Final Report JPL PO# NQ-802429, March 26: 52 pp.

Stewart, A. I. and W. B. Hanson (1982). Mars' Upper Atmosphere: Mean and Variations. Adv. Space Res., 2: 87-101

Sutton, J. L., C. B. Leovy and J. E. Tillman (1978). J. Atmos. Sci., 35: 2346-2355

Tillman, J. E., et al. (1979). Frontal Systems During Passage of the Martian North Polar Hood Over the Viking 2 Site Prior to the First 1977 Dust Storm. J. Geophys. Res., 84(B6): 2947-2955

Tillman, J. E. (1988). Mars Global Atmospheric Oscillations: Annually Synchronized, Transient Normal-Mode Oscillations and the Triggering of Global Dust Storms. J. Geophys. Res., 93(D8): 9433-9451

Toon, O. B., J. B. Pollack and C. Sagan (1977). Physical Properties of Particles Comprising the Martian Dust Storm, 1971-1972. Icarus, 30: 663-696

Zurek, R. W. (1976). Diurnal Tide in the Martian Atmosphere. J. Atmos. Sci., 33: 321-337

Zurek, R. W. (1982). Martian Great Dust Storms: An Update. Icarus, 50: 288-310

Zurek, R. W. (1986). Atmospheric Tidal Forcing of the Zonal-Mean Circulation: The Martian Dusty Atmosphere. J. Atmos. Sci., 43(7): 652-670

Zurek, R. W. (1988). Free and Forced Modes in the Martian Atmosphere. J. Geophys. Res., 93(D8): 9452-9462

Zurek, R. W. and R. M. Haberle (1988). Zonally Symmetric Response to Atmospheric Tidal Forcing in the Dusty Martian Atmosphere. J. Atmos. Sci., 45(18): 2469-2479

Page
B-20
Blank

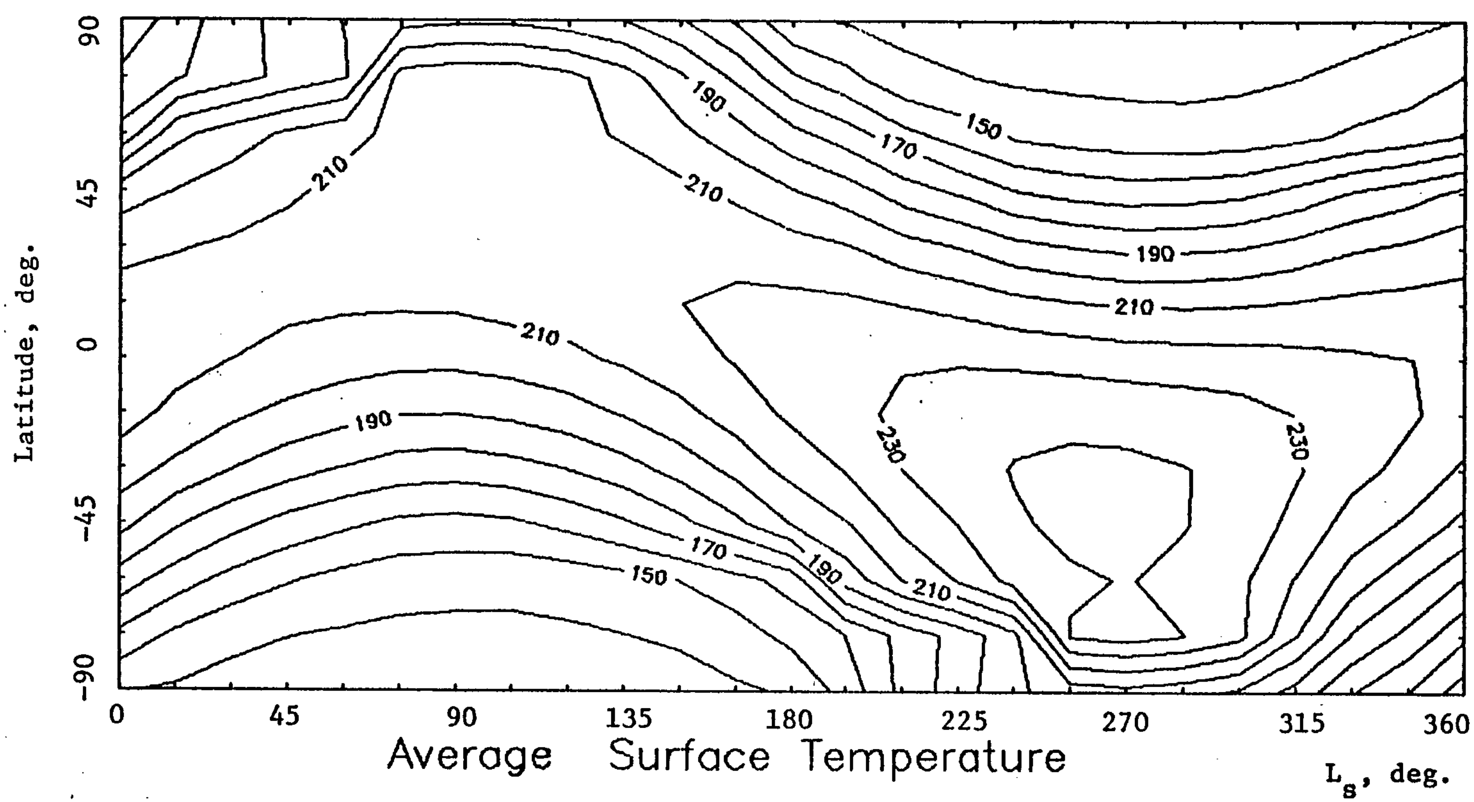


Figure 1 - Seasonal and latitudinal variation of daily average surface temperature, computed by the MARS-GRAM model. L_s is the areocentric longitude of the sun.

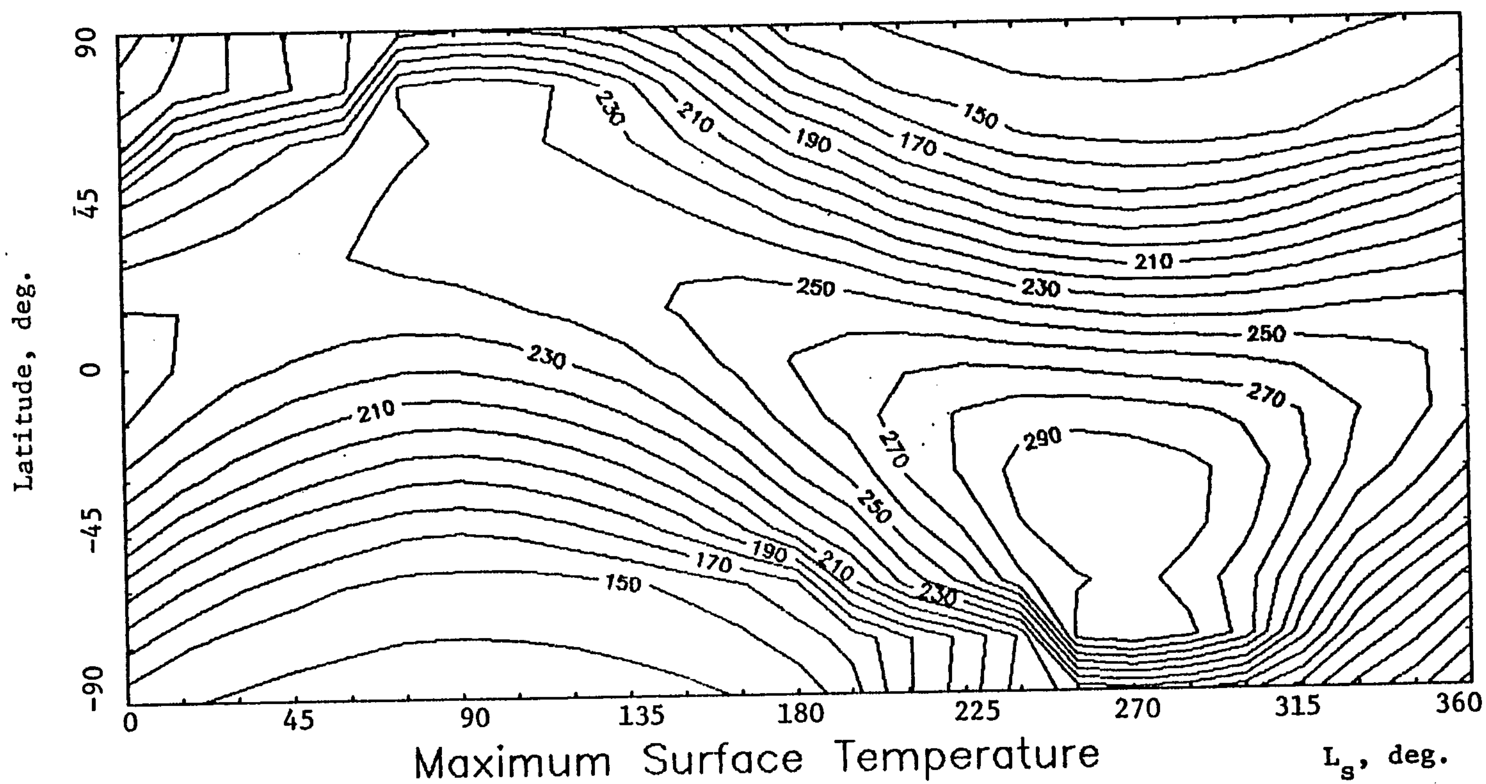


Figure 2 - Seasonal and latitudinal variation of daily maximum surface temperature, computed by the MARS-GRAM model.

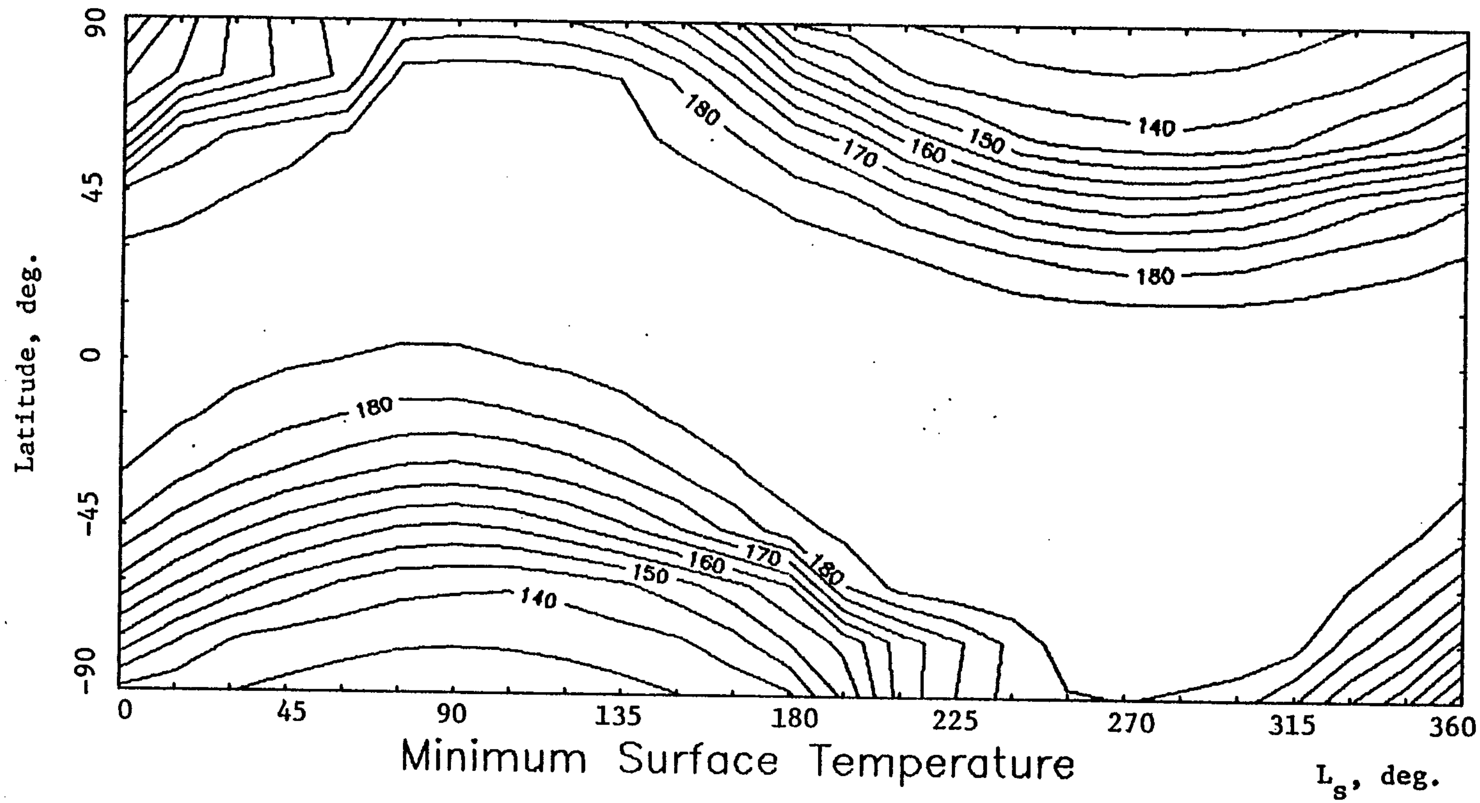
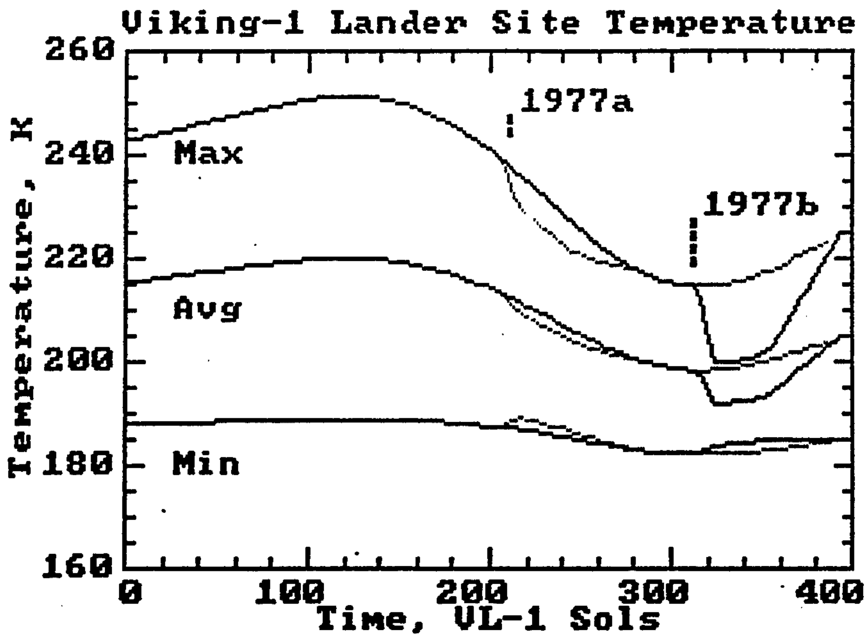
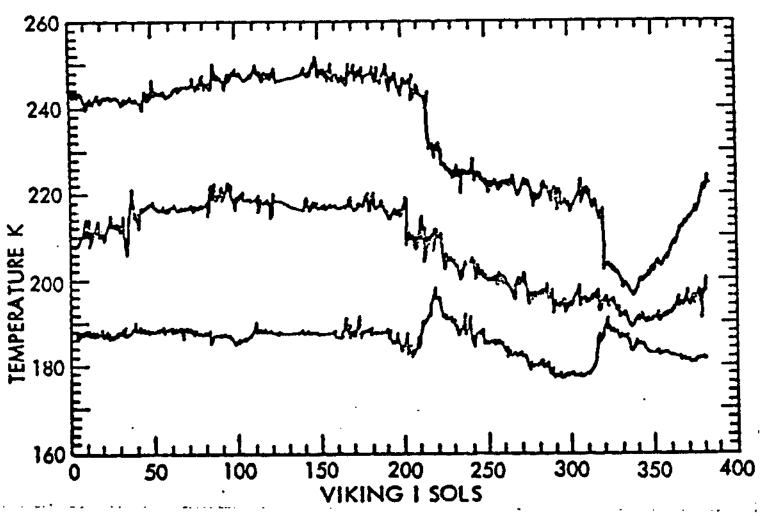


Figure 3 - Seasonal and latitudinal variation of daily minimum surface temperature, computed by the MARS-GRAM model.

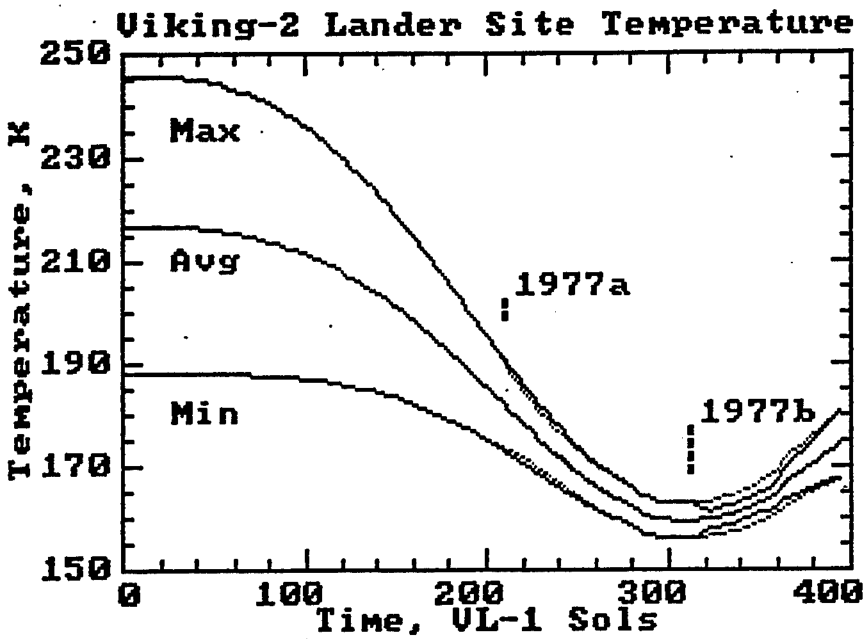


a)

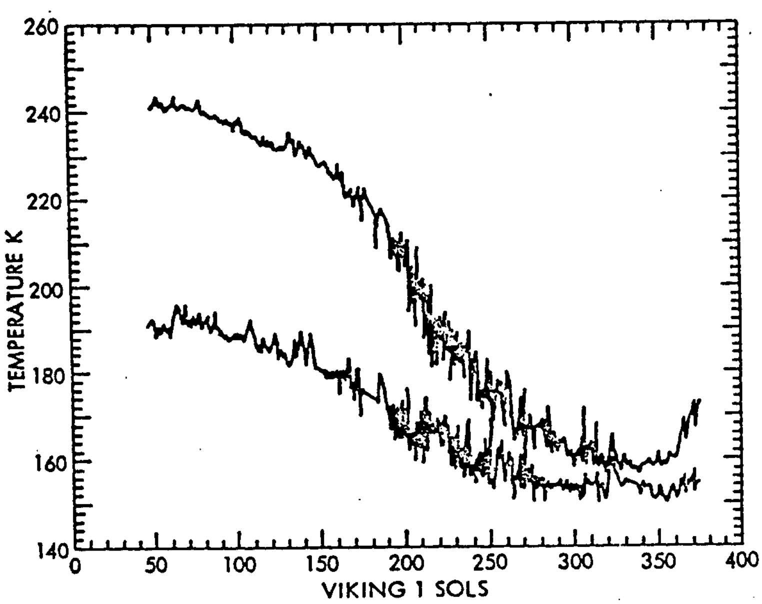


b)

Figure 4 - Seasonal variation of the daily maximum, mean, and minimum temperature at the Viking Lander 1 site (a) computed by MARS-GRAM, and (b) as reported by Ryan and Henry (1979).

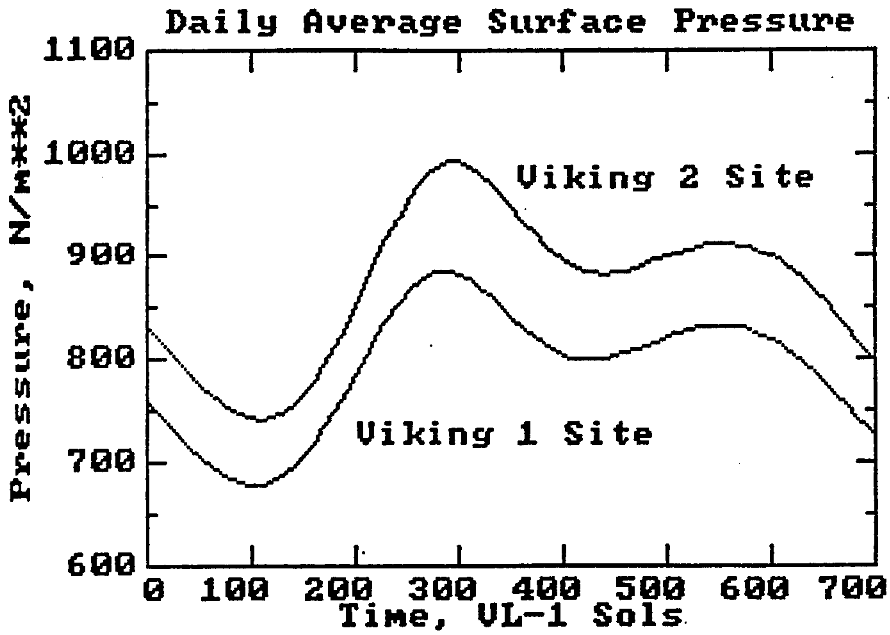


a)

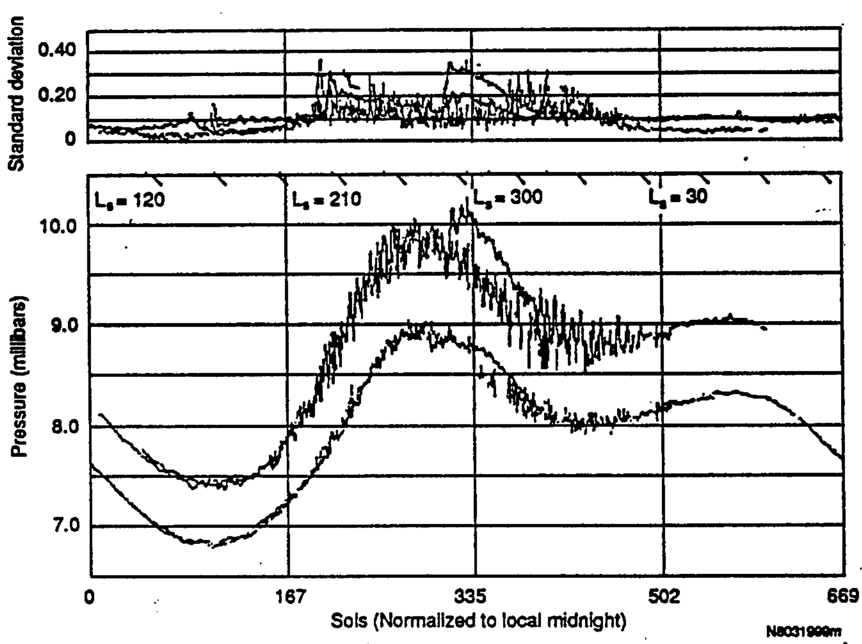


b)

Figure 5 - Seasonal variation of the daily maximum, mean, and minimum temperature at the Viking Lander 2 site (a) computed by MARS-GRAM, and (b) as reported by Ryan and Henry (1979).



a)



b)

Figure 6 - Seasonal variation of the daily average surface pressure at the Viking 1 and 2 Lander sites (a) computed by Mars-GRAM, and (b) as reported by Tillman (1988).

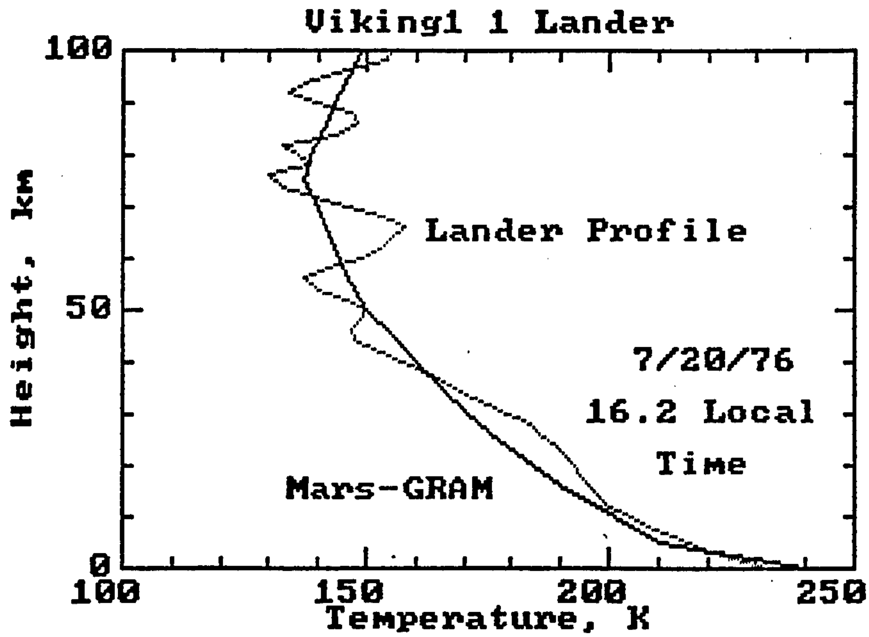


Figure 7 - Vertical temperature profile simulated by Mars-GRAM for date, time and position of Viking 1 Lander site (solid line) and measured Viking 1 Lander profile (dotted line).

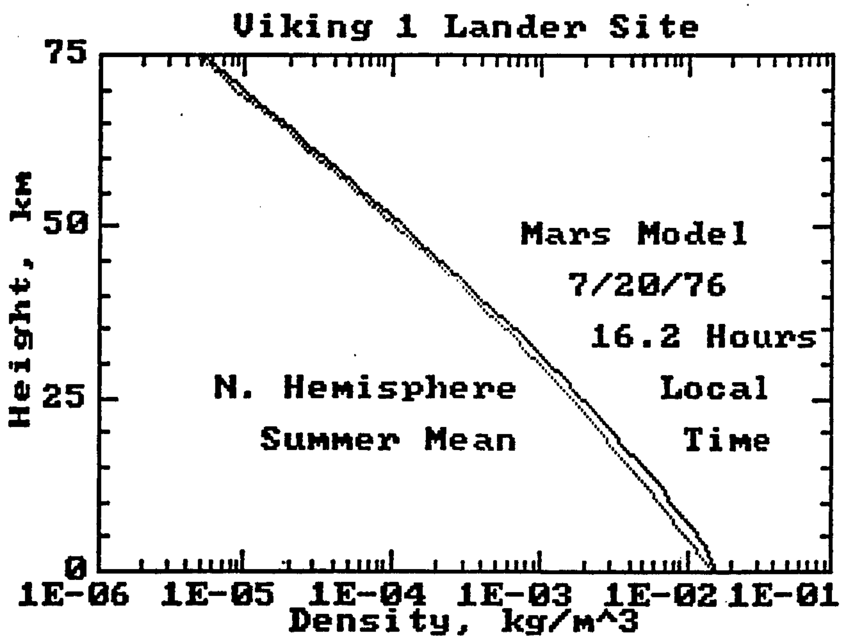


Figure 8 - Vertical profile of density simulated by the Mars-GRAM for date, time and position of the Viking 1 Lander site (solid line) and the COSPAR model for Northern Hemisphere summer mean conditions (dotted line).

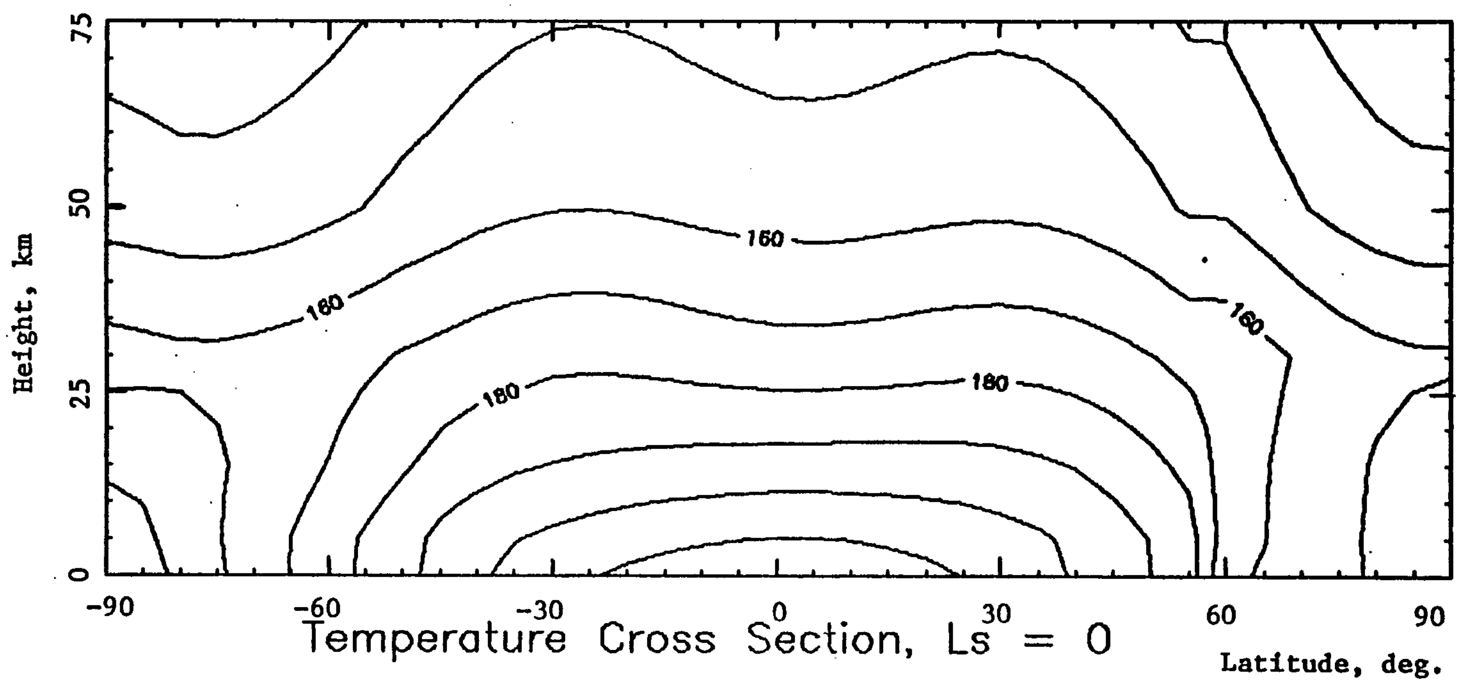


Figure 9 - Height and latitudinal variation of daily mean temperature at Northern Hemisphere spring Equinox (areocentric longitude of sun $L_s = 0°$) for dust-free conditions, as computed by Mars-GRAM.

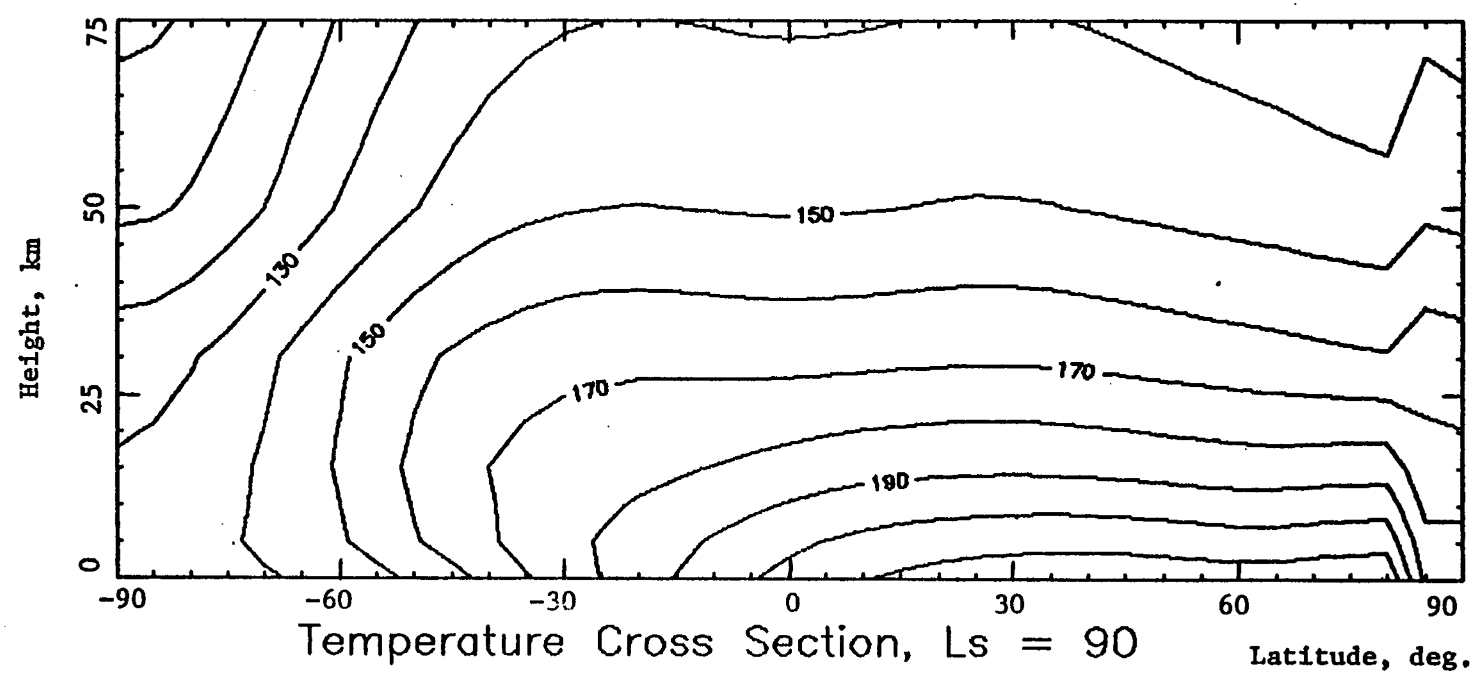


Figure 10 - Height and latitudinal variation of daily mean temperature at Northern Hemisphere summer solstice (L_s = 90°) for dust-free conditions, as computed by Mars-GRAM.

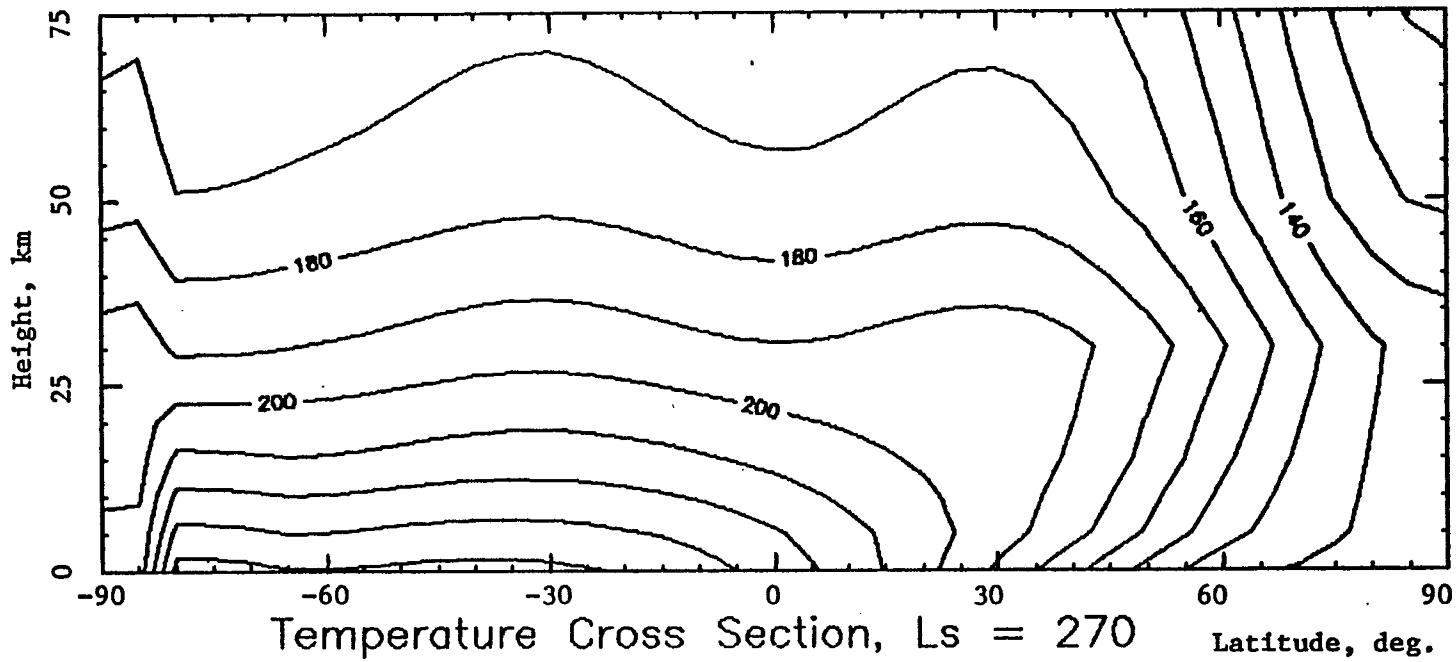


Figure 11 - Height and latitudinal variation of daily mean temperature at Northern Hemisphere winter solstice (L_s = 270°) for dust-free conditions, computed by Mars-GRAM.

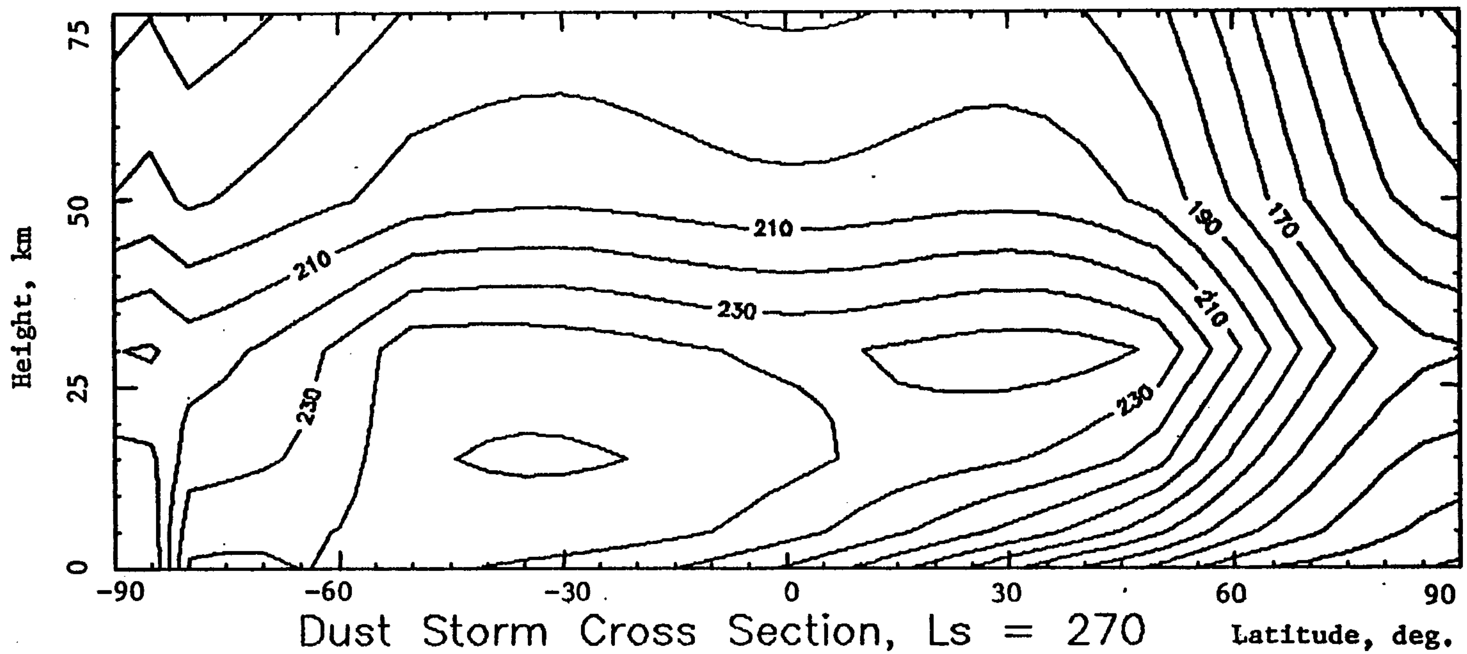
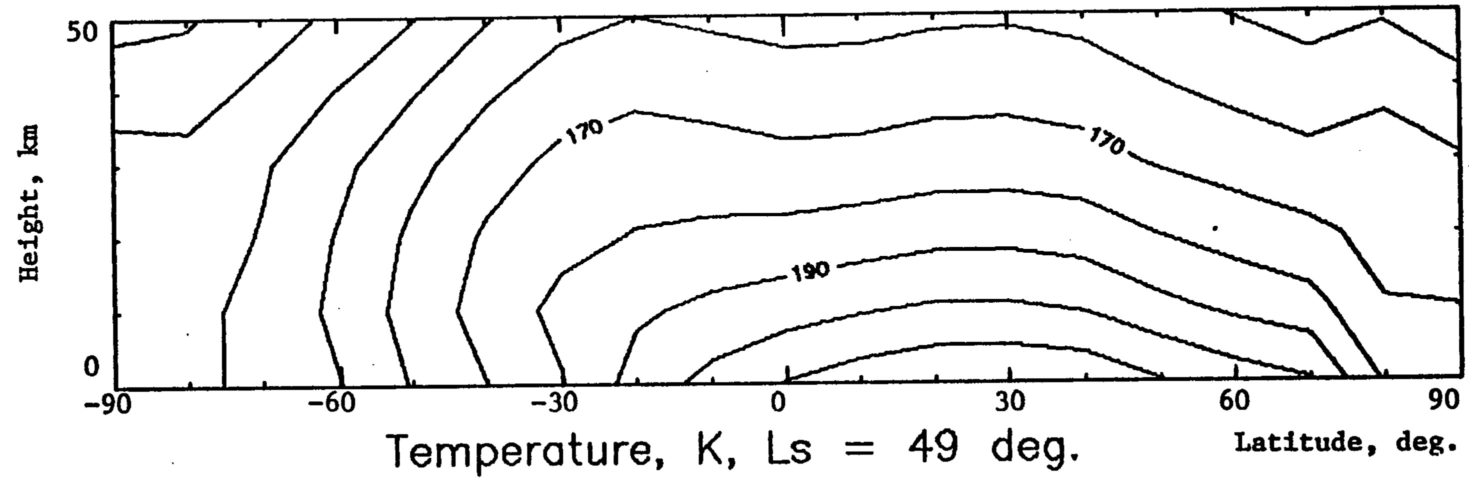
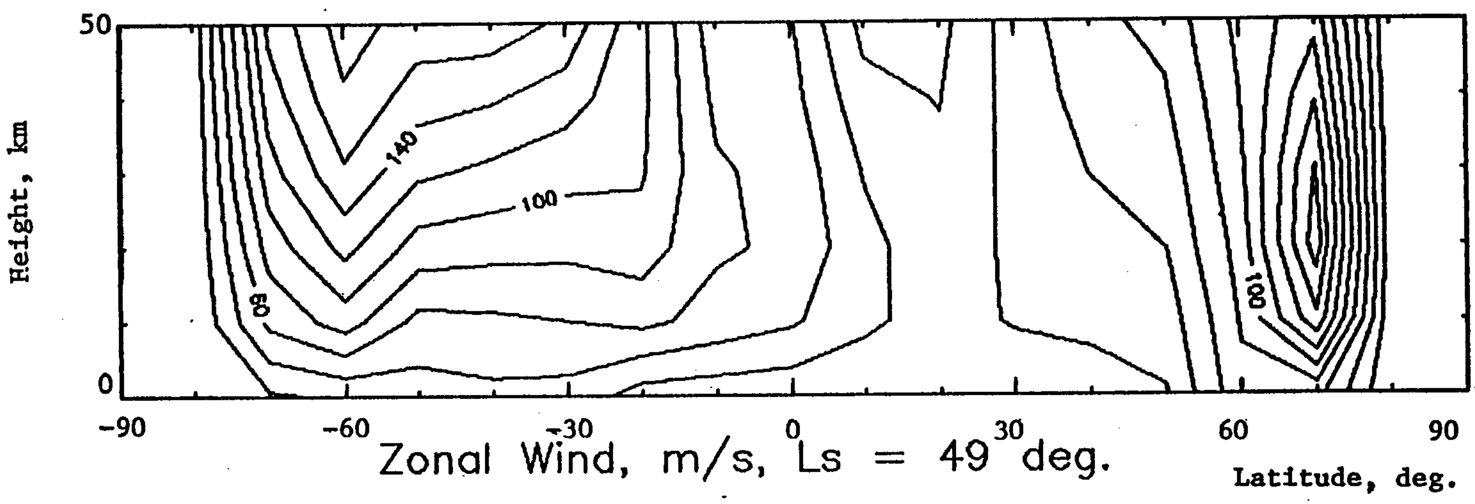


Figure 12 - Height and latitudinal variation of daily mean temperature at Northern Hemisphere winter solstice for a fully-developed global dust storm (intensity 3.0), computed by Mars-GRAM.



a)



b)

Figure 13 - Cross section of (a) temperature (K) and (b) zonal wind (m/s) at 9 am local time for $L_s = 49°$.

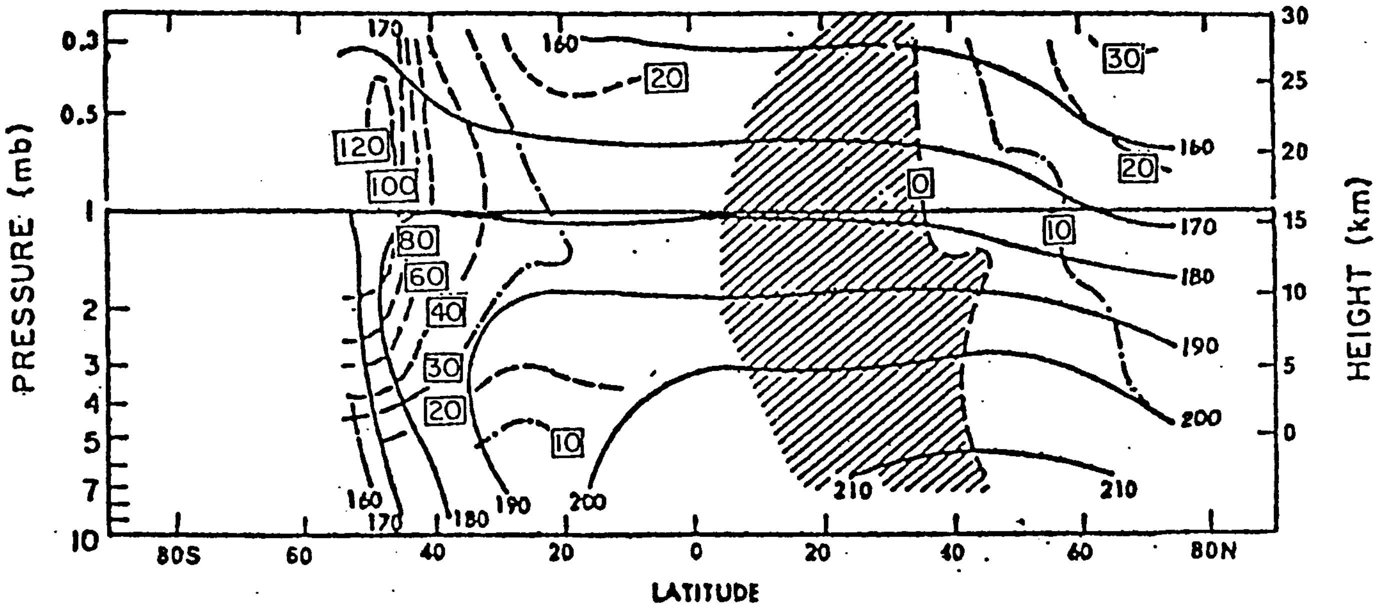
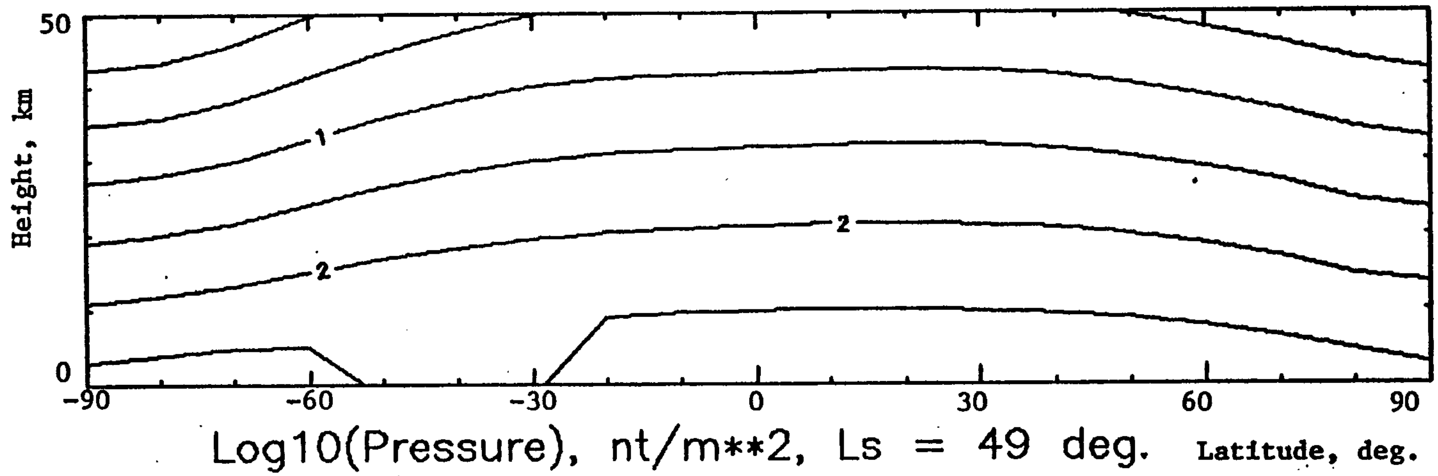
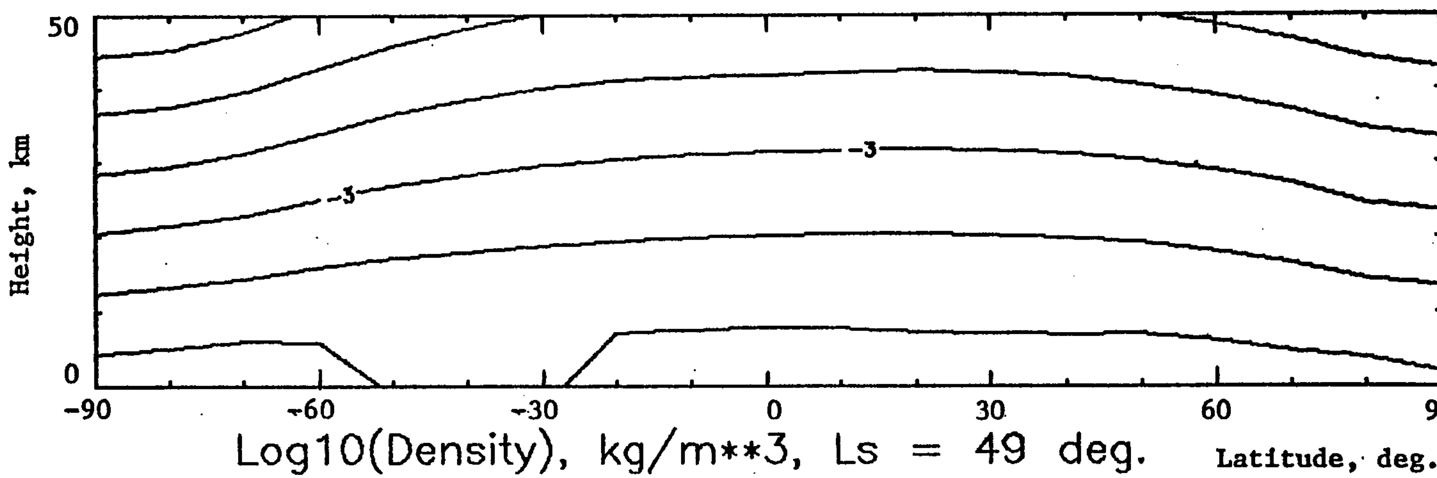


Figure 14 - Cross section of zonally-averaged temperature (K) and geostrophic zonal wind (m s^{-1}) based on Mariner 9 IRIS data between L_s = 43 and L_s = 54 (approximately early May in analogous terrestrial season), Leovy (1982), from data provided by B. Conrath.



a)



b)

Figure 15 - Cross sections of (a) log-base-10 pressure (N/m^2) and (b) log-base-10 density (kg/m^3) at 9 am local time for $L_s = 49°$.

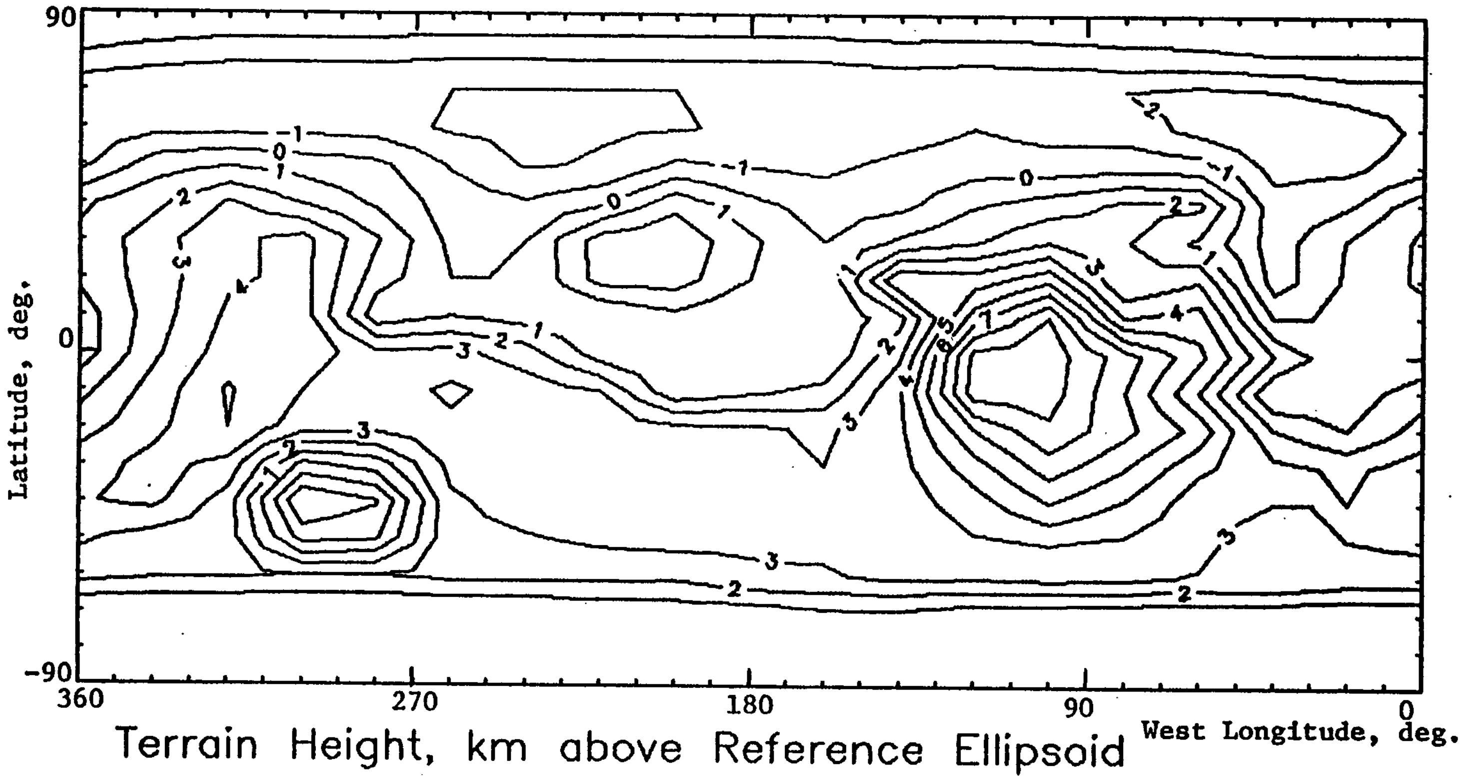


Figure 16 - Contours of local surface terrain height, relative to reference ellipsoid (km) at 10° latitude by 20° longitude resolution.

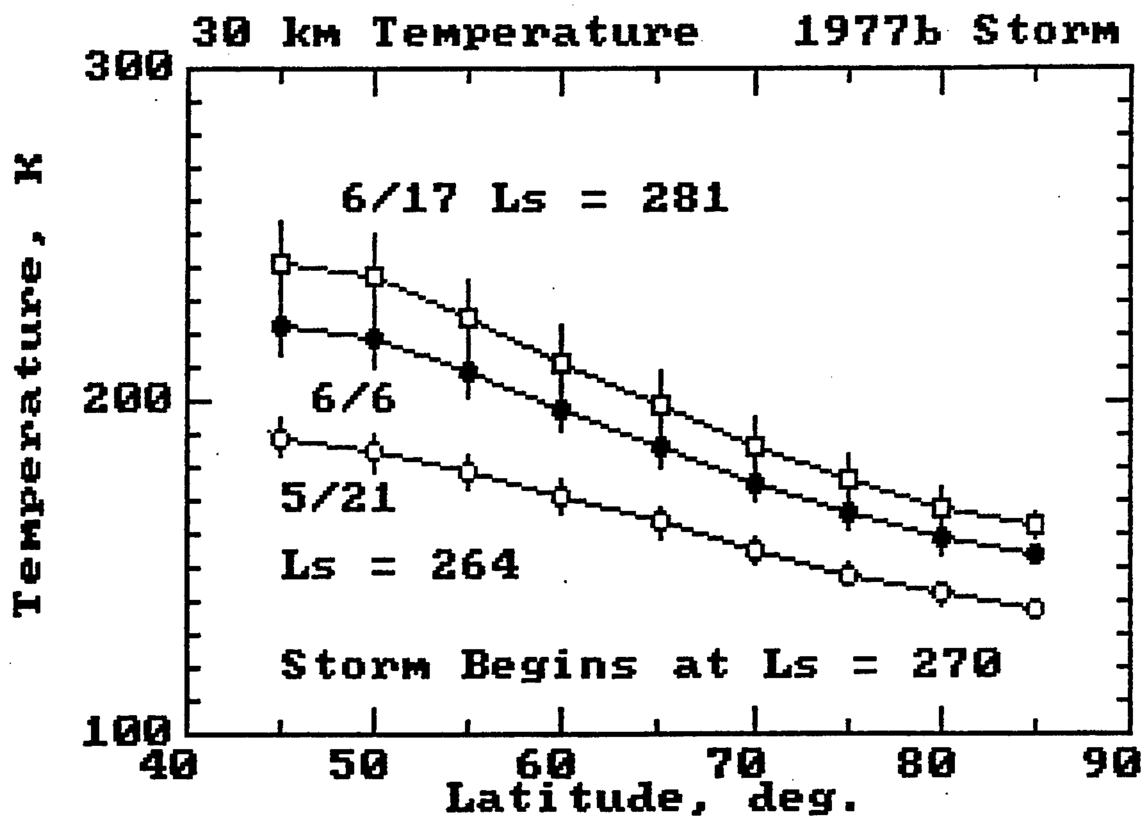


Figure 17 - Progression of simulated dust-storm effect on daily average, maximum and minimum temperature versus latitude and L_s value (degrees) for the 1977b storm.

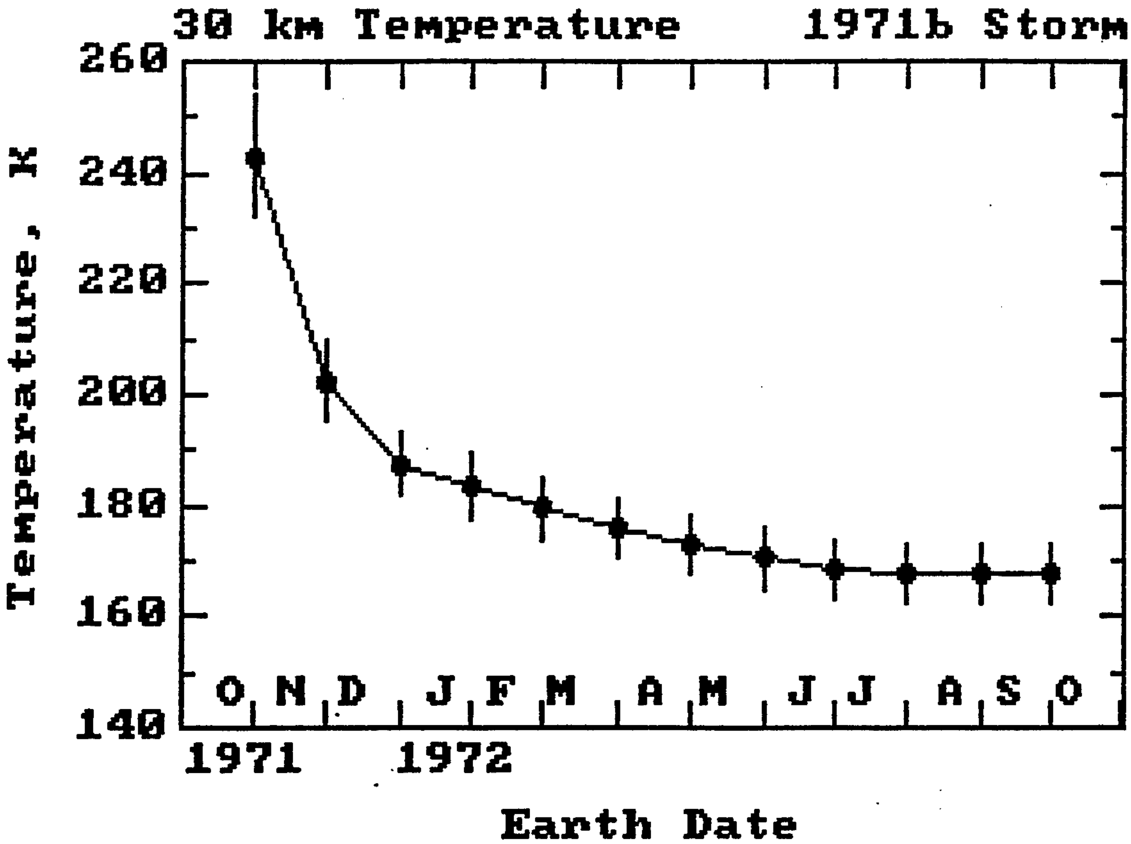


Figure 18 - Progression of simulated dust-storm effect on daily average, maximum and minimum temperature versus time at latitude 25°S for the 1971b storm.

APPENDIX C

MARS-GRAM RELEASE #2 TECHNICAL REPORT

This Appendix contains the technical portions of the Release #2 Report for Mars-GRAM (version 3.1), "The Mars Global Reference Atmospheric Model (Mars-GRAM) Release #2", Bonnie F. James (Grant Monitor) and C. G. Justus, March 1, 1993, prepared under Georgia Tech grant NAG8-877 for NASA Marshall Space Flight Center. Appendix material (giving outdated information on program input, output, and running characteristics) has been deleted.

ABSTRACT

Three major additions or modifications have been made to the Mars Global Reference Atmospheric Model (Mars-GRAM). (1) In addition to the interactive version, a new batch version is available, which uses NAMELIST input, and is completely modular, so that the main driver program can easily be replaced by any calling program, such as a trajectory simulation program. (2) Both the interactive and batch versions now have an option for treating local-scale dust storm effects, rather than just the global-scale dust storms in the original Mars-GRAM. (3) The Zurek wave perturbation model has been added, to simulate the effects of tidal perturbations, in addition to the random (mountain wave) perturbation model of the original Mars-GRAM. A minor modification has also been made which allows heights to go "below" local terrain height and return "realistic" pressure, density and temperature, not the surface values, as returned by the original Mars-GRAM. This feature will allow simulations of Mars rover paths which might go into local "valley" areas which lie below the average height of the present, rather coarse-resolution, terrain height data used by Mars-GRAM. Sample input and output of both the interactive and batch versions of Mars-GRAM are presented.

INTRODUCTION

The Mars Global Reference Atmospheric Model (Mars-GRAM; Justus and Chimonas, 1988; Justus, 1990, 1991), includes simulation capabilities for mean values of density, temperature, pressure and wind components. Density perturbations, simulated by a mountain-wave simulation model are also included, for simulation of density perturbation profiles along specified trajectories through the atmosphere of Mars.

Up to a height of 75 km, Mars-GRAM is based upon parameterizations of height, latitudinal, longitudinal and seasonal variations of temperature, determined from a survey of published measurements from the Mariner and Viking programs (a complete set of references is provided by Justus and Chimonas, 1989). Pressure and density are inferred in Mars-GRAM from the temperature by making use of the hydrostatic and perfect gas law relationships. Above about 120 km, Mars-GRAM uses the thermospheric model of Stewart (1987). A hydrostatic interpolation routine is used to insure a smooth transition from the lower portion of the model to the Stewart thermosphere model.

Mars-GRAM includes parameterizations to simulate the effects of seasonal variation, diurnal variation, dust storm effects, effects due to the orbital position of Mars, effects of the large seasonal variation in surface atmospheric pressure because of differential condensation/sublimation of the CO_2 atmosphere in the polar caps, and effects of Martian atmospheric mountain wave perturbations on the magnitude of the expected density perturbations. The thermospheric model includes a parameterization for the effects of solar activity, measured by the 10.7 cm solar radio flux. Winds are computed by the thermal wind approximation, with the inclusion of the effects of molecular viscosity, which, because of the low atmospheric densities, can be very important at high altitudes. The mountain wave perturbation model also includes a damping approximation due to the effects of molecular viscosity.

During much of 1990 and early 1991, the Mars Atmosphere Knowledge Requirements Working Group (Bourke, 1991) met by video-conference, with the objectives of

- Understanding the sensitivity and consequences of various levels of uncertainties in the Martian atmosphere

- Understanding the realistic limits on modeling the Mars atmosphere, both deterministically and statistically

- Recommending a set of atmospheric information requirements to be satisfied by the robotic portion of the Mars Exploration Program

Specific recommendations were made by the working group for improvements and additions to the Mars-GRAM program:

(1) The addition of a capability to treat local-scale dust storms. The current Mars-GRAM treats all storms as growing and decaying with time, but with a size of full global dimensions.

(2) The addition of the Zurek wave model (Pitts et al., 1988) to represent the large-scale temperature and density perturbations caused by atmospheric tides. The current Mars-GRAM perturbations are rather small-scale, gravity-wave-like variations.

(3) A modular version of Mars-GRAM, specifically designed for use as a subroutine in a calling program, such as a trajectory guidance and control analysis program. This version would allow easy application of user-defined perturbation models, such as sine-wave perturbations or hyperbolic tangent or step-like perturbations.

The purpose of this report is to describe the Mars-GRAM modifications recently made to satisfy these recommendations.

ADDITIONS TO MARS-GRAM

The Zurek Wave Perturbation Model

Parameters necessary to evaluate wave-structure perturbations in temperature are provided by Zurek's tables on pages 11-12 of Pitts et al. (1988). The Zurek wave model was designed to allow estimation of temperature perturbations which would be produced by atmospheric tides (expected to be an important process in the Martian atmosphere). In order to compute wave-structure perturbations in density from the original temperature perturbation estimates of Zurek, a simple model is assumed, based on approximate hydrostatic equilibrium for these large-scale perturbations. The perfect gas law, $p = \rho RT$, requires that perturbations in pressure, density and temperature be related (to first order) by

$$p'/\langle p\rangle \;=\; \rho'/\langle \rho\rangle \;+\; T'/\langle T\rangle \qquad , \tag{1}$$

where the angle brackets denote average values. The perturbation version of the hydrostatic equations, $\partial p'/\partial z = -\rho' g$, requires that

$$\partial(p'/\langle p\rangle)\partial z = -(\rho'/\langle\rho\rangle - p'/\langle p\rangle)\,(g/RT) \;=\; -(T'/\langle T\rangle)\,(g/RT) \qquad . \tag{2}$$

If one assumes a simple cosine function for the vertical variation of $T'/\langle T\rangle$, i.e.,

$$T'/\langle T\rangle \;=\; A\cos(kz) \qquad , \tag{3}$$

then equations (1) and (2) require that

$$\rho'/\langle\rho\rangle \;=\; A\,[\sin(kz)\,/\,(kH) - \cos(kz)] \qquad , \tag{4}$$

where H is the scale height (RT/g). The temperature perturbation data provided in Zurek's table provide information to determine the values of k and the wave amplitude A for both clear-sky and dust-storm cases. If it is assumed that $kH \cong 1$, then the density perturbation may be evaluated from this k value, with the approximation form of equation (4), namely

$$\rho'/\langle\rho\rangle \cong A\,[\sin(kz) \;-\; \cos(kz)] \qquad . \qquad (5)$$

Figure 1 shows a sample evaluation of the Zurek density wave perturbation model at the location and time of the Viking 1 lander entry (7/20/76 12:30 GMT, at Mars latitude 22°N, longitude 48°W). Since the random (mountain-wave) perturbations no longer constitute the only perturbations in Mars-GRAM, their minimum acceptable perturbation magnitudes have been decreased somewhat. A sample of the random density perturbations at the Viking 1 lander location and time is shown in Figure 2. The total density perturbations, found by adding the wave-component and random-component perturbations from Figures 1 and 2, are consistent with the density perturbations observed by the Viking 1 and 2 landers (shown as figures in the Pitts et al. 1988 report), which have peak values of 10-20%.

Figure 3 shows the density wave perturbation values from the Zurek model, evaluated on a height-latitude cross section, through the Viking 1 longitude (48°N) at the time of the lander entry. Peak contour values in Figure 3 are +14% and -14% (not labeled because of the small area within these contours).

Local-Scale Dust Storm Simulations

At run time, the Mars-GRAM user selects the time of start (within seasonal bounds) for a dust storm (if any) to be simulated. A time profile of build-up and decay for the dust storm intensity (up to a selectable maximum value) is part of the program. The new additions also allow selection of a location (latitude and longitude) at which the dust storm is to start and a maximum radius (up to 10,000 km) that the dust storm is allowed to grow. These parameters of dust-storm location and maximum radius (r_{max}) are used to compute a size factor which multiplies the intensity of the dust storm effects. The size factor, as a function of position and height, is given by

$$\text{size factor} \;=\; 0.25\,[1 + \cos(90°r/r_d)][1 + \cos(90°z/z_d)]\;, \qquad (6)$$

where r is the local radius from the dust storm center location, r_d is the temporally varying dust-storm radius (up to a maximum of r_{max}), z is the local height and z_d is the height of the dust storm. z_d also grows temporally up to a maximum value of 60 km or $r_d/3$ (whichever is smaller).

The dust storm radius r_d and height z_d are values for 1/2 the full effect. Thus the size factor given by equation (6) is 1 when $r/r_d = 0$ and $z/z_d = 0$; it is 1/2 when $r/r_d = 1$ and $z/z_d = 1$; and it is 0 when $r/r_d = 2$ or $z/z_d = 2$. As illustrated by Figure 4 the function $0.5[1 + \cos(x)]$ used in equation (6) is very similar to the Gaussian distribution function $\exp(-x^2/\pi)$ frequently used as a size factor function is diffusion models.

Figure 5 illustrates the effects of a local dust storm on the mean density simulated along a hypothetical trajectory, starting at latitude 0°N, longitude 0°W at 40 km altitude, and moving at constant height and latitude along longitude from 0 to 100°W, with a local dust storm centered at latitude 0°N, longitude 50°W, with radius = 1000 km. The values plotted in Figure 5 are the differences between mean density with dust storm perturbation and that with no dust storm effects.

Deviations in mean density of 25% or more are seen near the center position of the local storm, with 0 deviations seen at distances from the storm center of more than twice the storm radius.

The Mars-GRAM Batch Version

A new batch version of Mars-GRAM has been developed, which uses NAMELIST input, and is completely modular, so that the main driver program can easily be replaced by any calling program, such as a trajectory simulation program. As with the interactive version, fixed values of trajectory displacements in height, latitude, longitude and time may be read in as input, or position along an arbitrary trajectory may be read in from an input file. In the batch version of Mars-GRAM, values of the trajectory displacement values can easily be changed with time by simple modifications to the short driver program, or from within the trajectory program which replaces the Mars-GRAM batch version driver program.

ADDITIONS TO MARS-GRAM

The wave perturbations are modeled in a new subroutine WAVEPERT and associated functions AMPRINT and PHASINT to interpolate wave amplitudes and phases. The Zurek wave model parameters are input to the program via a new BLOCK DATA routine. The local dust storm modifications are incorporated into the previous subroutine DUSTFACT.

Other program modifications are that:

(1) A minor modification has also been made which allows heights to go "below" local terrain height and return "realistic" pressure, density and temperature, not the surface values, as returned by the original Mars-GRAM. This feature will allow simulations of Mars rover paths which might go into local "valley" areas which lie below the average height of the present, rather coarse-resolution, terrain height data used by Mars-GRAM.

(2) For simulations which are to follow the Mars-GRAM terrain heights exactly, an input height below -5 km will specify this option.

(3) For 1-D plots versus either of the height variables (height above reference ellipsoid or above local terrain), the plotable output files have the height variable in the second (y) position (ordinate). This simplifies input to plot routines which do not allow run-time selection of which input variable is the abcissa and which is the ordinate.

DIGITAL TERRAIN DATA

A new set of digital terrain height data for Mars, produced by the U.S. Geological Survey, has been received on large format magnetic tape. The feasibility of extracting a more detailed set of terrain heights from this set, for improvement over those currently used by Mars-GRAM, has been examined. Some potential problems with this are that the data on DEC VAX format tapes (we have no easy access to a DEC VAX with 9-track tape drives), and the data on the tapes are in "pixel" form for ease in producing images, not necessarily for extracting tabular data at an array (e.g., 1° X 1°) of fixed locations.

This digital terrain model data base is now available on CD-ROM (Batson, 1992), and has been ordered in this format. However, the data were not received in time to process during this contract period.

REFERENCES

Batson, R. M., et al. (1992). CD-ROM Publication of the Mars Digital Cartographic Data Base, NTIS HC/MF A25.

Bourke, R. D., editor (1991). Report of the Mars Atmosphere Knowledge Requirements Working Group, JPL Technology Report.

Justus, C. G. (1990). A Mars Global Reference Atmospheric Model (Mars-GRAM) for Mission Planning and Analysis, AIAA 90-004, presented at the 28th Aerospace Sciences Meeting, Reno, NV, January.

Justus, C. G. (1991). Mars Global Reference Atmospheric Model for Mission Planning and Analysis. J. Spacecraft and Rockets, 28(2): 216-221

Justus, C. G. and G. Chimonas (1989). The Mars Global Reference Atmospheric Model (Mars-GRAM). NASA MSFC Technical Report, ORIG 7-20-89, REV1 10-8-89.

Pitts, D. E., J. E. Tillman, J. Pollack and R. Zurek (1988). Model Profiles of the Mars Atmosphere for the Mars Rover and Sample Return Mission, draft technical report, March 11.

Stewart, A. I. F. (1987). Revised Time Dependent Model of the Martian Atmosphere for use in Orbit Lifetime and Sustenance Studies. Final Report JPL PO# NQ-802429, March 26.

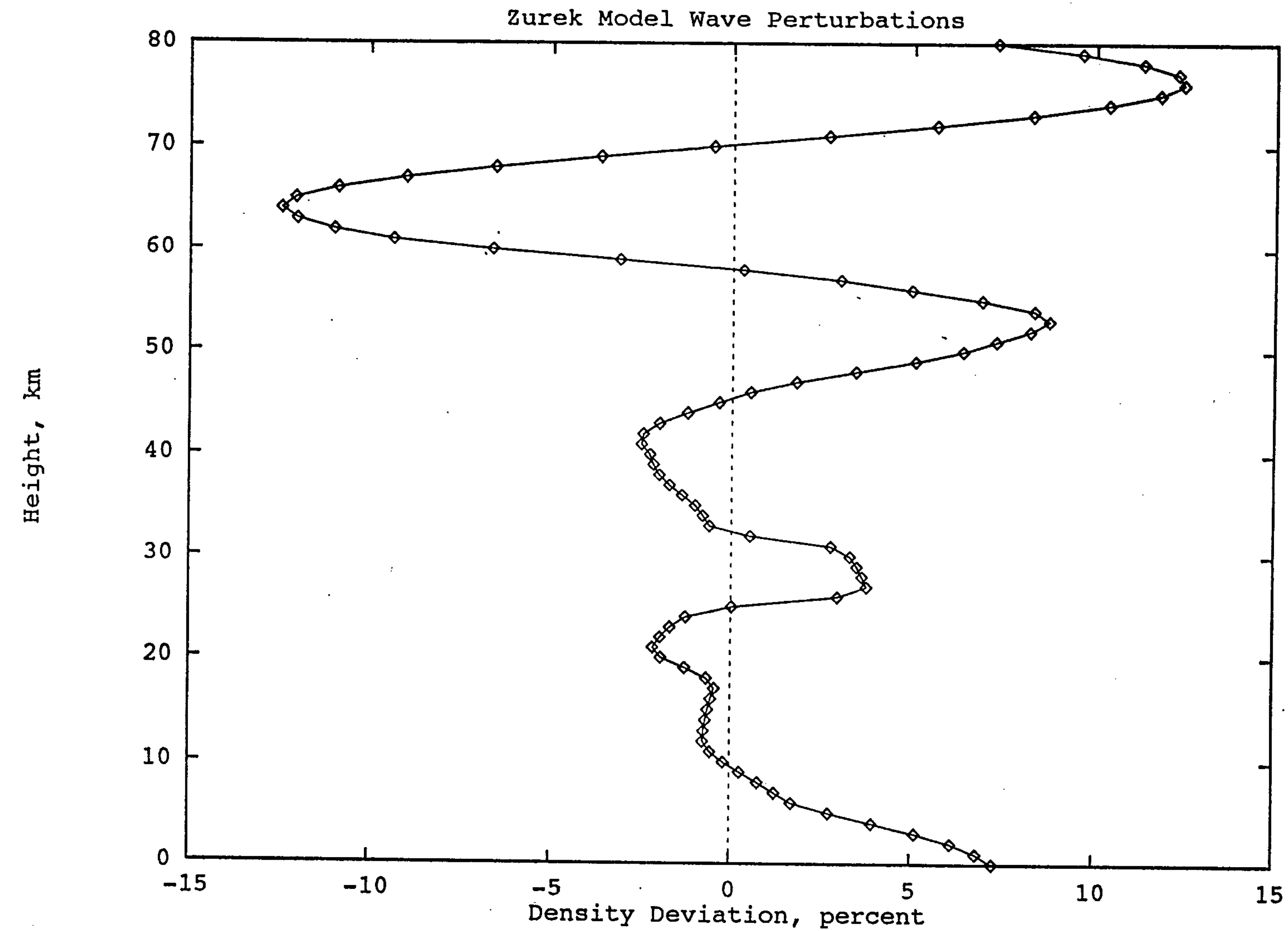


Figure 1 - Sample evaluation of the Zurek density wave perturbation model (based on temperature perturbation parameters in Pitts et al., 1988). Location and time corresponds to that of the Viking 1 Lander entry (22°N 48°W, 12:30 GMT on 7/20/76).

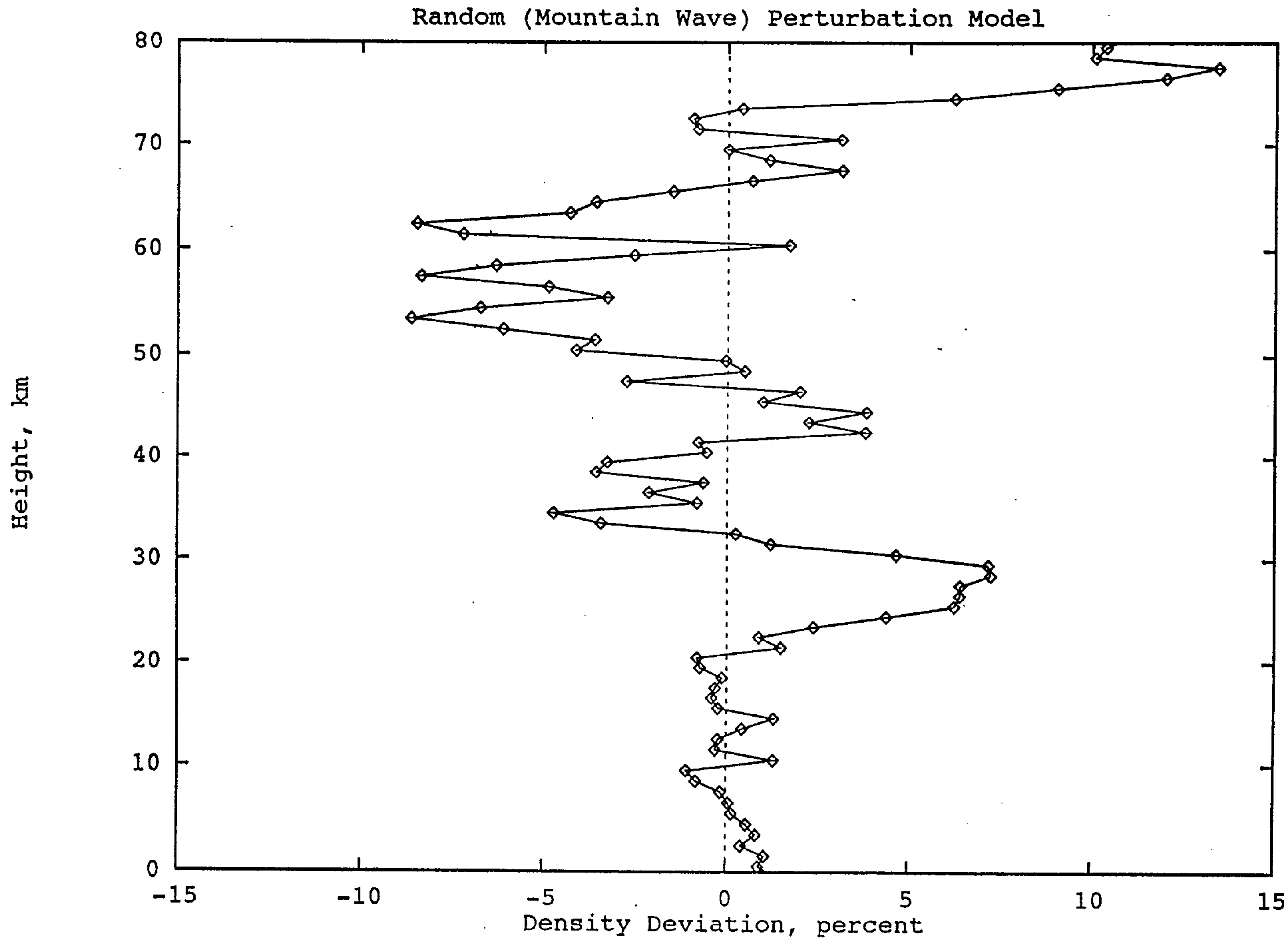


Figure 2 - Random (mountain-wave) perturbations in density, evaluated for the same location and time as in Figure 1.

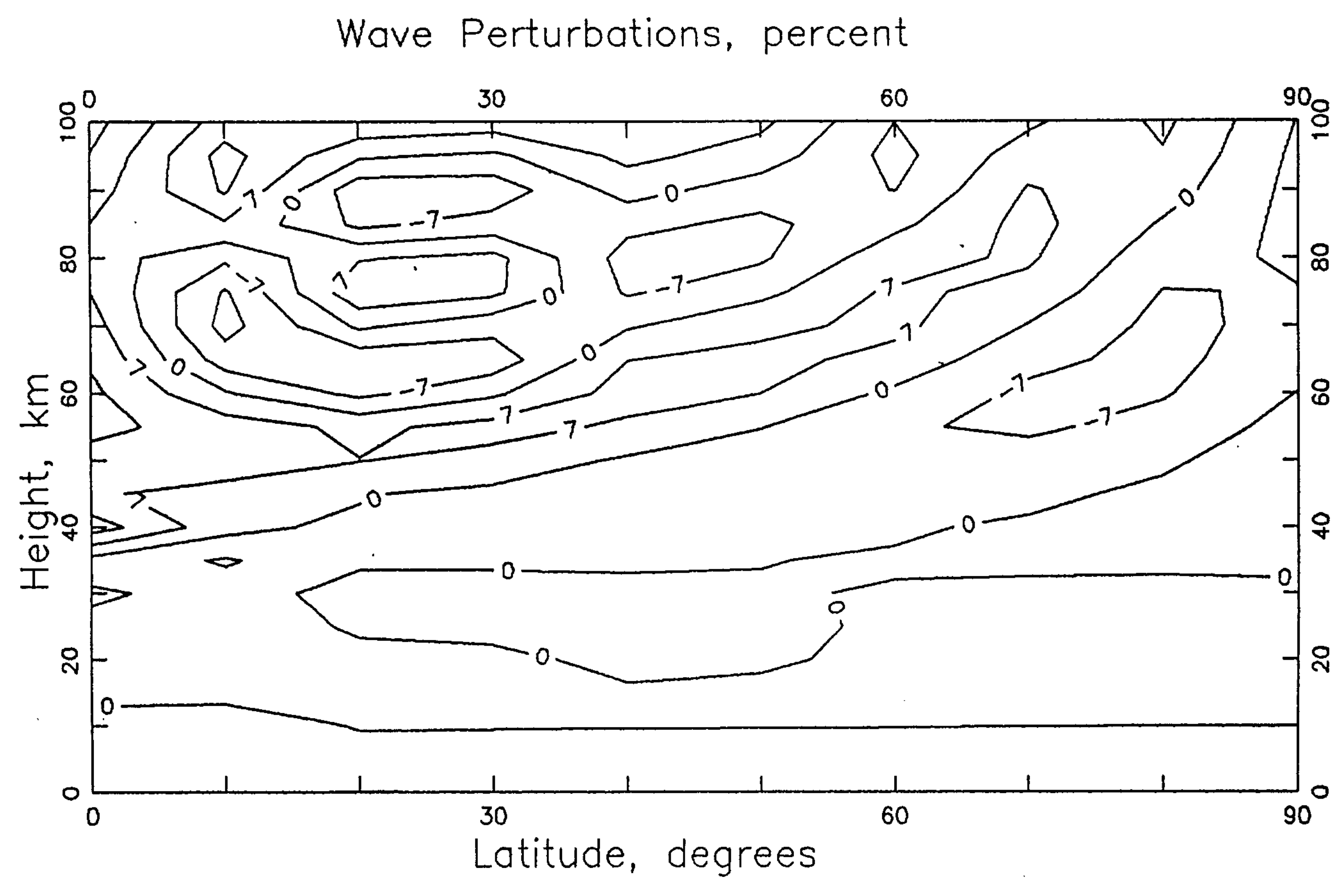


Figure 3 - Zurek density wave perturbations evaluated along a height-latitude cross section through the longitude 48°W at the time of the Viking 1 Lander entry.

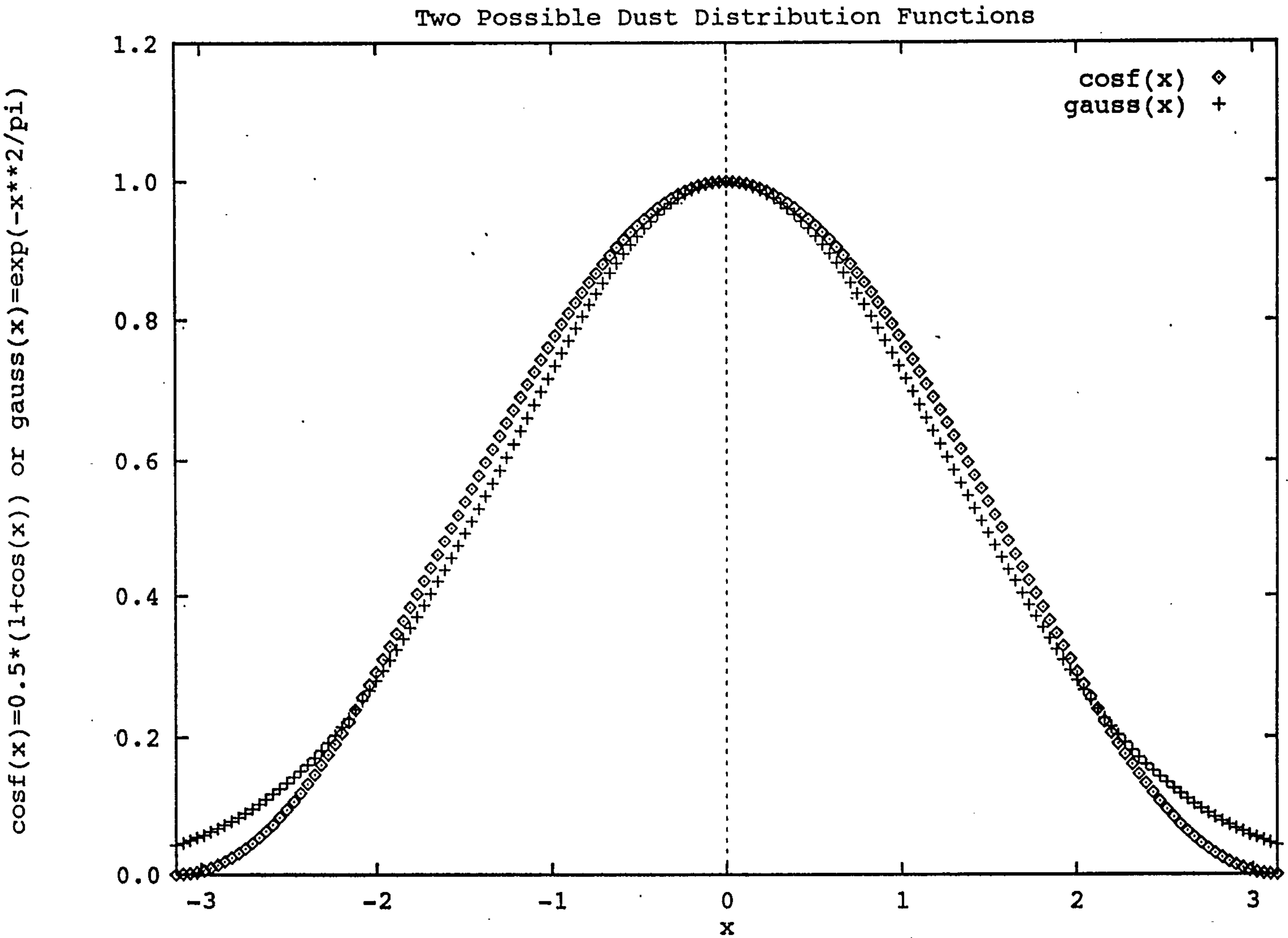


Figure 4 - Comparison of two possible spatial distribution model functions for local-scale dust storms. $cosf(x) = 0.5[1 + \cos(x)]$, for $-\pi \le x \le \pi$, with x in radians, and $gauss(x) = \exp(-x^2/\pi)$. The functions are normalized to 1 at x=0, with the area under cosf(x) from $-\pi \le x \le \pi$ the same as the area under gauss(x) from $-\infty \le x \le +\infty$.

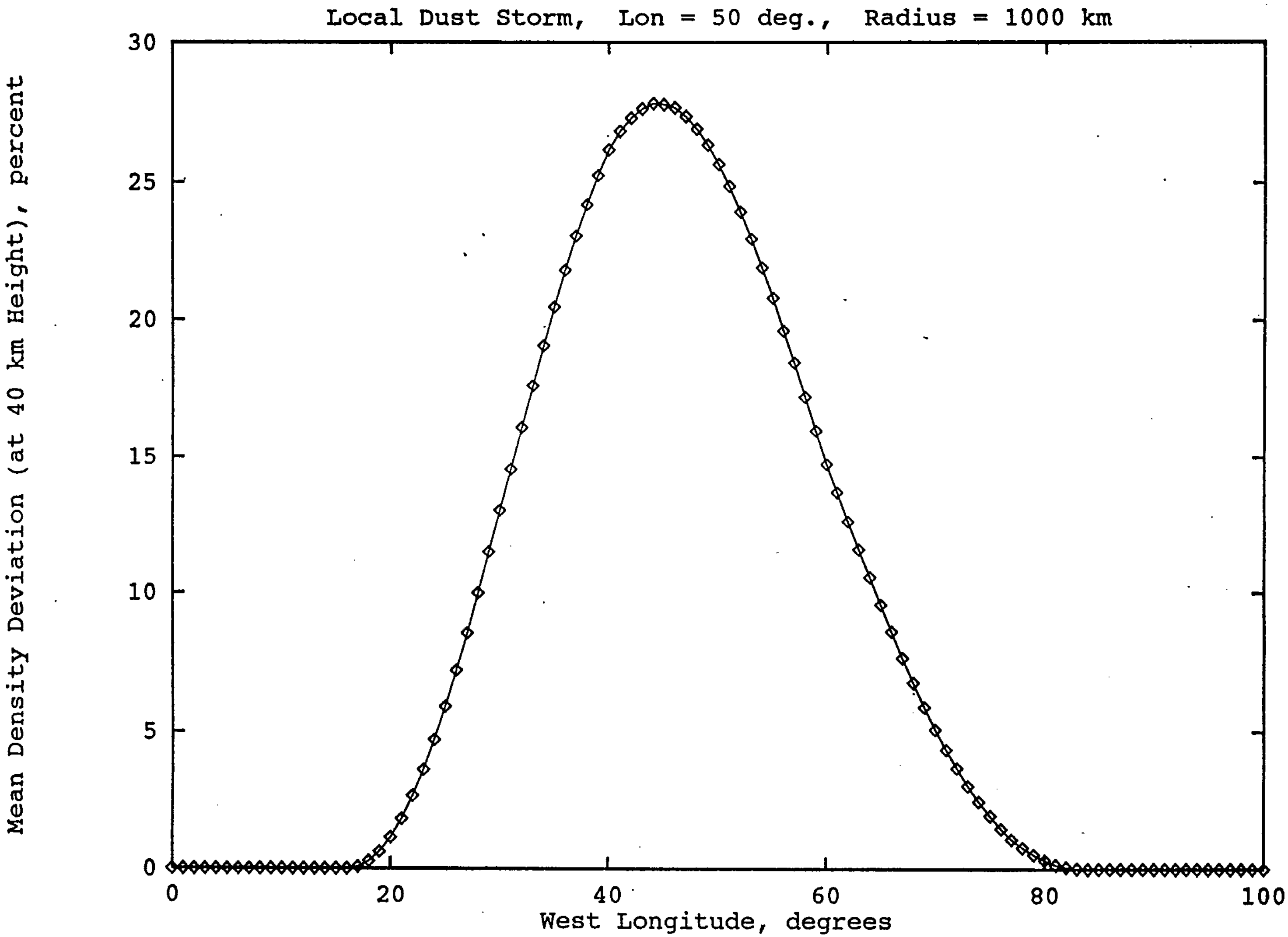


Figure 5 - The effect on mean atmospheric density due to a local dust storm of radius 1000 km, centered at 0°N, 50°W. Density was evaluated along a trajectory at a constant height of 40 km from 0°N, 0°W to 0°N, 100°W. Mean density deviations are in percent, relative to density in the non-dust storm case.

APPROVAL

MARS GLOBAL REFERENCE ATMOSPHERIC MODEL (MARS-GRAM 3.34): PROGRAMMER'S GUIDE

C. G. Justus, Bonnie F. James and Dale L. Johnson

The Information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classifications Officer. This report, in its entirety, has been determined to be unclassified.

James N. Strickland
Director, Systems Analysis and Integration Laboratory

Robert E. Smith
Chief, Systems Engineering Division

Steven D. Pearson
Chief, Electromagnetics and Aerospace Environments Branch

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, Va 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	May 1996	Technical Memorandum

4. TITLE AND SUBTITLE

Mars Global Reference Atmospheric Model (Mars-GRAM 3.34): Programmer's Guide

5. FUNDING NUMBERS

6. AUTHOR(S)

C.G. Justus*, B.F. James and D.L. Johnson

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

8. PERFORMING ORGANIZATON REPORT NUMBERS

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA TM-108509

11. SUPPLEMENTARY NOTES

Prepared by Electromagnetics and Aerospace Environments Branch, System Analysis and Integration Laboratory, Science and Engineering Directorate.
*Computer Sciences Corporation, Huntsville, Alabama

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified - Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This is a programmer's guide for the Mars Global Reference Atmospheric Model (Mars-GRAM 3.34). Included are a brief history and review of the model since its origin in 1988 and a technical discussion of recent additions and modifications. Examples of how to run both the interactive and batch (subroutine) forms are presented. Instructions are provided on how to customize output of the model for various parameters of the Mars atmosphere. Detailed descriptions are given of the main driver programs, subroutines, and associated computational methods. Lists and descriptions include input, output, and local variables in the programs. These descriptions give a summary of program steps and "map" of calling relationships among the subroutines. Definitions are provided for the variables passed between subroutines through "common" lists. Explanations are provided for all diagnostic and progress messages generated during execution of the program. A brief outline of future plans for Mars-GRAM is also presented.

14. SUBJECT TERMS

Mars Global Reference Atmospheric Model, thermosphere, atmospheric temperature, pressure, wind components, atmospheric models

15. NUMBER OF PAGES

189

16. PRICE CODE

NTIS

17. SECURITY CLASSIFICATION	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

NSN 7540 - 01 - 280 - 5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std. 239-18

National Aeronautics and
Space Administration
Code JTT
Washington, DC
20546–0001

Official Business
Penalty for Private Use, $300

Postmaster: If Undeliverable (Section 158 Postal Manual), Do Not Return